Some decidability results on one-pass reductions

Sándor Vágvölgyi *
Department of Foundations of Computer Science
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
e-mail: vagvolgy@inf.u-szeged.hu

Abstract

We show second-order decidability results on recognizable tree languages and one-pass reduction sequences with special term rewriting systems. For some classes of term rewriting systems the second-order IO one-pass inclusion problem, the second-order IO one-pass reachability problem, and the IO and OI one-pass common ancestor problems are decidable.

Keywords: term rewriting systems; IO and OI one-pass reductions; tree automata

1 Introduction

One-pass algorithms have been studied and implemented in many areas of computer science, for example in image processing, data stream processing, and database systems. Dauchet and De Comité [4], Seynhaeve et al. [19] studied inside-out (IO) and outside-in (OI) one-pass reductions for term rewriting systems (TRSs for short). Along an IO one-pass reduction sequence we proceed from the leaves to the root, and along an OI one-pass reduction sequence we proceed from the root to the leaves. All reduction steps can be carried out mainly independently from each other. During a reduction step, the left-hand side of an applied term rewriting rule does not overlap with the already rewritten parts of the term, only the values of the substituted subterms depend on the order of the reduction steps. Along an IO one-pass we substitute already rewritten subterms, along an OI one-pass we replace subterms of the initial input term. Fülöp et al. [7] studied two other very restrictive strategies of term rewriting: one-pass leaf-started rewriting and one-pass root-started rewriting. They differ from the IO and OI one-pass reductions, respectively, in that the rewriting is always applied at positions immediately adjacent to the already rewritten parts of the term. Consequently, they establish a much more restricted way of computing.

Reachability is a fundamental problem that appears in several areas of computer science: finite- and infinite-state concurrent systems, computational models like cellular automata and Petri nets, program analysis, discrete and continuous systems, time critical systems, hybrid systems, TRSs, etc. [3, 18, 20, 22]. Important research has been carried out on reachability of TRSs [2, 5, 6, 10, 11, 12, 14, 15, 16, 17, 21]. Ground reachability and unreachability proofs for TRSs can be used as general purpose verification techniques for the systems modeled by rewriting [10]. Gilleron and Tison [13], Fülöp et al. [7], and Seynhaeve et al. [19] introduced and studied various second-order reachability and inclusion problems for various types of reductions, they replaced individual trees in the original problem by recognizable tree languages. In the light of the above problems, Vágvölgyi [23] studied the following eight second-order decidability problems. The terms appearing in an IO (resp. OI) one-pass reduction sequence are called the IO (resp. OI) one-pass sentential forms of the initial term. For a tree language L, the set of all IO (resp. OI) one-pass sentential forms of the elements of L

*This research was supported by the NKFI grant no K 108448 and the project "The Hu-Maths-In Deepening the activities of the Hungarian Industrial and Innovation Mathematics Services Network", no EFOP-3.6.2-16-2017-00015. The project has been supported by the European Union and co-funded by the European Social Fund.

Table 1: Already known decidability results

decidability of	inclusion	reachability	joinability	common ances.
IO one-pass reduction	decidable for ll [19, Prop. 4]	?	?	undecidable for rl [23, Theorem 3]
OI one-pass reduction	decidable for rl [19, Prop. 4] decidable for ll [23, Theorem 1]	decidable for ll [23, Theorem 1]	undecidable for ll [23, Theorem 2]	?

is denoted by $IOSF(L)$ (resp. $OISF(L)$). First we present the problems concerning IO one-pass reducing [23].

Second-order IO one-pass inclusion problem.
Instance: A TRS R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is $IOSF(L) \subseteq M$?

Second-order IO one-pass reachability problem.
Instance: A TRS R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is $IOSF(L) \cap M \neq \emptyset$?

Second-order IO one-pass joinability problem.
Instance: A TRS R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is $IOSF(L) \cap IOSF(M) \neq \emptyset$?

Second-order IO one-pass common ancestor problem.
Instance: A TRS R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is there a term $t \in T_\Sigma(X)$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$?

One defines the OI counterparts of the above problems replacing IO by OI. The concept of a TRS without the variable restrictions (TRS_+ for short) is obtained by dropping the variable restrictions in the definition of a TRS. We generalize the above second order decidability problems for TRS_+s replacing TRS by TRS_+.

We present the decidability results on one-pass reductions with TRSs in the literature in Table 1, where ll, rl, ances., and wa abbreviate left-linear, right-linear, ancestor, and weakly ambiguous respectively. Each question mark signifies an open problem. We sum up the already known results and our results in Table 2. Here we write + to indicate that a result concern TRS_+s rather than TRSs.

In Section 2, we present our notations and basic definitions. We say that a TRS_+ R is right weakly overlapping (rwo) if the following condition holds: for any two right-hand sides r_1, r_2 of term rewriting rules in R, and for any nonvariable subterm r_3 of r_2 with $r_3 \neq r_2$, r_1, r_3 do not have a common instance, furthermore if r_1, r_2 have a common instance, then r_1 is equal to r_2 up to renaming of variables. We say that a TRS_+ R keeps the non-linear variables of the left-hand sides (R is knlv) if for any term rewriting rule $l \to r$ in R, each non-linear variable of l appears in r. In Section 3, we present our results on IO one-pass reduction sequences with knlv rwo TRS_+s rather than TRSs to simplify the proofs of Section 4 on OI one-pass reductions. For any right-linear knlv rwo TRS_+ R and recognizable tree language L over Σ, we can construct a tree automaton $\mathcal{B}$ recognizing $IOSF(L)$. Therefore, for right-linear knlv rwo TRS_+s, the second-order IO one-pass joinability problem is decidable. For right-linear knlv rwo TRS_+s the second-order IO one-pass inclusion problem and the second-order IO one-pass reachability problem are decidable. For left-linear TRS_+s, the second-order IO one-pass common ancestor problem is decidable.

Table 2: Summary of already known and this paper's decidability results

decidability of	inclusion	reachability	joinability	common ances.
IO one-pass reduction	decidable for ll [19, Prop. 4] decidable for rl knlv rwo + Theorem 3.14	decidable for rl knlv rwo + Theorem 3.14	decidable for rl knlv rwo + Corollary 3.13	undecidable for rl [23, Theorem 3] decidable for ll Corollary 3.29
OI one-pass reduction	decidable for rl [19, Prop. 4] decidable for ll [23, Theorem 1]	decidable for ll [23, Theorem 1]	undecidable for ll [23, Theorem 2]	decidable for ll wa Corollary 4.5

We say that a TRS_+ R is weakly ambiguous if the following condition holds: for any two left-hand sides l_1, l_2 of term rewriting rules in R, l_1 and any nonvariable subterm l_3 of l_2 with $l_3 \neq l_2$, l_1 and l_3 do not have a common instance, furthermore if l_1, l_2 have a common instance, then l_1 is equal to l_2 up to renaming of variables. In Section 4, we show that for left-linear weakly ambiguous TRSs, the second-order OI one-pass common ancestor problem is decidable. We derive this result from its IO counterpart, which states that for right-linear knlv rwo TRS_+s, the second-order IO one-pass joinability problem is decidable.

In Section 5, we present our concluding remarks and open problems.

2 Preliminaries

We recall and invent some notations, basic definitions and terminology which will be used in the rest of the paper. Nevertheless the reader is assumed to be familiar with the basic concepts of term rewriting systems and of tree language theory [1, 8, 9].

2.1 Sets and words

Let $\to \subseteq A \times A$ be a binary relation on a set A. We denote by $\to^*$ the reflexive, transitive closure of $\to$. We denote the set of all nonempty subsets of a set S by $P_+(S)$. The set of nonnegative integers is denoted by $\mathbb{N}$, and $\mathbb{N}^*$ stands for the free monoid generated by $\mathbb{N}$ with empty word λ as identity element. Consider the words $\alpha, \beta, \gamma \in \mathbb{N}^*$ such that $\alpha = \beta\gamma$. Then we say that β is a prefix of α, and that α is an extension of β; and we write $\beta \preceq \alpha$. If $\gamma \neq \lambda$, then β is a proper prefix of α, and we write $\beta \prec \alpha$. For any $\alpha, \beta, \in \mathbb{N}^*$, we say that α and β are incomparable, if α is not a prefix of β and β is not a prefix of α.

2.2 Terms

A ranked alphabet is a finite set Σ in which every symbol has a unique rank in $\mathbb{N}$. For $m \geq 0$, Σ_m denotes the set of all elements of Σ which have rank m. The elements of Σ_0 are called constants. Throughout the paper we assume that $\Sigma_0 \neq \emptyset$. That is, we have at least one constant in Σ.

For a set of variables Y and a ranked alphabet Σ, $T_\Sigma(Y)$ denotes the set of Σ-terms (or Σ-trees) over Y. $T_\Sigma(\emptyset)$ is written as T_Σ. A term $t \in T_\Sigma$ is called a ground term. A tree language L is a subset of T_Σ. A tree $t \in T_\Sigma(Y)$ is linear if any variable of Y occurs at most once in t. A variable $y \in Y$ appearing at least twice in $t \in T_\Sigma(Y)$ is called a non-linear variable of t. We specify a countable set $X = \{ x_1, x_2, \dots \}$ of variables which will be kept fixed in this paper. Moreover, we put

$X_m = \{ x_1, \ldots, x_m \}$, for $m \geq 0$. Hence $X_0 = \emptyset$. Let $t \in T_\Sigma(X)$. Then $var(t) \subseteq X$ denotes the set of variables occurring in t.

For a term $t \in T_\Sigma(X)$, the set of positions $POS(t) \subseteq \mathbb{N}^*$ of t is defined by tree induction.

- If $t \in \Sigma_0 \cup X$, then $POS(t) = \{ \lambda \}$.
- If $t = f(t_1, \ldots, t_m)$ with $f \in \Sigma_m,\ m > 0$, then
  $POS(t) = \{ \lambda \} \cup \{ i\alpha \mid 1 \leq i \leq m, \alpha \in POS(t_i) \}$.

For every $t \in T_\Sigma(X)$ and $\alpha \in POS(t)$, we introduce the subterm $t/\alpha \in T_\Sigma(X)$ of t at α and define the label $lab(t, \alpha) \in \Sigma \cup X$ in t at α as follows:

- for $t \in \Sigma_0 \cup X$, $t/\lambda = t$ and $lab(t, \lambda) = t$;
- for $t = f(t_1, \ldots, t_m)$ with $m \geq 1$ and $f \in \Sigma_m$, if $\alpha = \lambda$ then $t/\alpha = t$ and $lab(t, \alpha) = f$, otherwise, if $\alpha = i\beta$ with $1 \leq i \leq m$, then $t/\alpha = t_i/\beta$ and $lab(t, \alpha) = lab(t_i, \beta)$.

For every $t \in T_\Sigma(X)$, we call a position $\alpha \in POS(t)$ of t a variable position if $lab(t, \alpha) \in X$. The set of variable positions of t is denoted by $VPOS(t)$. That is, $VPOS(t) = \{ \alpha \in POS(t) \mid lab(t, \alpha) \in X \}$. For every $t \in T_\Sigma(X)$, $sub(t) = \{ t/\alpha \mid \alpha \in POS(t) \}$ is the set of all subtrees of t.

For every $k \geq 0$, we introduce the set $CO_\Sigma(X_k)$ of k-contexts as a subset of $T_\Sigma(X_k)$ as follows: $CO_\Sigma(X_k)$ consists of all trees $t \in T_\Sigma(X_k)$ such that the variable x_i appears exactly once in t for every $i \in \{ 1, \ldots, k \}$. For a context $u \in CO_\Sigma(X_k)$ and a variable $x_i \in X_k$, the address $adr(u, x_i)$ of x_i in u is the unique position $\alpha \in POS(u)$ such that $lab(u, \alpha) = x_i$.

For trees $t \in T_\Sigma(X)$ and $t_1, \ldots, t_m \in T_\Sigma(X)$ with $m \geq 0$, we denote by $t[t_1, \ldots, t_m]$ the tree obtained by substituting t_i for every occurrence of x_i in t, for $1 \leq i \leq m$. We call $t[t_1, \ldots, t_m]$ an instance of t. If a term $p \in T_\Sigma(X)$ is an instance of both $s \in T_\Sigma(X)$ and $t \in T_\Sigma(X)$, then we say that p is a common instance of s and t. For each term $t \in T_\Sigma(X_m)$ with $m \geq 0$, there is an integer $k \geq 0$ and a context $u \in C_\Sigma(X_k)$ such that $t = u[x_{i_1}, \ldots, x_{i_k}]$ for some $i_1, \ldots, i_k \in \{ 1, \ldots, m \}$. Then $var(t) = \{ x_{i_1}, \ldots, x_{i_k} \}$ and for any terms $t_1, \ldots, t_m \in T_\Sigma(X)$,

$$t[t_1, \ldots, t_m] = (u[x_{i_1}, \ldots, x_{i_k}])[t_1, \ldots, t_m] = u[t_{i_1}, \ldots, t_{i_k}].$$

We call an injective map $\phi : X \to X$ a variable renaming. For any two terms $s, t \in T_\Sigma(X)$, we say that s is equal to t up to renaming of variables, and write $s \simeq t$, if there is a variable renaming $\phi : X \to X$ such that t is obtained from s by substituting $\phi(x_i)$ for every occurrence of x_i in s for each $x_i \in var(s)$.

For $t \in T_\Sigma$, $\alpha \in POS(t)$, and $r \in T_\Sigma$, we define $t[\alpha \leftarrow r] \in T_\Sigma$ as follows.

- If $\alpha = \lambda$, then $t[\alpha \leftarrow r] = r$.
- If $\alpha = i\beta$ and $t = f(t_1, \ldots, t_m)$ for some $i \in \mathbb{N}$, $\beta \in \mathbb{N}^*$, and $f \in \Sigma_m$ with $1 \leq i \leq m$, then $t[\alpha \leftarrow r] = f(t_1, \ldots, t_{i-1}, t_i[\beta \leftarrow r], t_{i+1}, \ldots, t_m)$.

2.3 Term Rewriting Systems

A term rewriting system (TRS) R over a ranked alphabet Σ is a finite subset of $(T_\Sigma(X) - X) \times T_\Sigma(X)$ such that for every $(l, r) \in R$, every variable of r also occurs in l. A term rewriting system without the variable restrictions (TRS_+) R over a ranked alphabet Σ is a finite subset of $T_\Sigma(X) \times T_\Sigma(X)$. Here we allow $(l, r) \in R$ with $l \in X$ and we do not require that for every $(l, r) \in R$, every variable of r also should occur in l. Obviously, each TRS R is a TRS_+ as well. Hence all concepts we shall define for TRS_+ in the sequel, are also defined for TRSs. Elements (l, r) of a TRS_+ R are called term rewriting rules and are denoted by $l \to r$. We call l the left-hand side and r the right-hand of the term rewriting rule $l \to r$. The set of left-hand sides (resp. right-hand sides) of term rewriting rules in R is denoted by $lhs(R)$ (resp. $rhs(R)$). The converse of a TRS_+ R is defined as $R^{-1} = \{ r \to l \mid l \to r \in R \}$. Obviously R^{-1} is also a TRS_+ over Σ. A TRS_+ R is left-linear (resp. right-linear) if every element of $lhs(R)$ (resp. $rhs(R)$) is linear. A left-linear and right-linear TRS_+ R is called linear. A TRS_+ R is ground if every element of $lhs(R) \cup rhs(R)$ is a ground term. A term rewriting rule $l \to r$ is collapsing if $r \in X$. A TRS$_+$ R is collapsing if it has a collapsing term rewriting rule , and non-collapsing otherwise. A set $S \subseteq T_\Sigma(X)$ of terms weakly overlaps if for any two terms $s, t \in S$,

- for all $\alpha \in (POS(s) - (\{\lambda\} \cup VPOS(s)))$, s/α and t have no common instance, and
- if s and t have a common instance, then $s \simeq t$.

A TRS$_+$ R is weakly ambiguous (wa) if $lhs(R)$ weakly overlaps. A TRS$_+$ R is right weakly overlapping (rwo) if $rhs(R)$ weakly overlaps. For the sake of clarity, we repeat the above definitions explicitly. We say that a TRS$_+$ R is wa if for any two left-hand sides $l_1, l_2 \in lhs(R)$,

- for all $\alpha \in (POS(l_1) - (\{\lambda\} \cup VPOS(l_1)))$, l_1/α and l_2 have no common instance, and
- if l_1 and l_2 have a common instance, then $l_1 \simeq l_2$.

We say that a TRS$_+$ R is rwo if for any two right-hand sides $r_1, r_2 \in rhs(R)$,

- for all $\alpha \in (POS(r_1) - (\{\lambda\} \cup VPOS(r_1)))$, r_1/α and r_2 have no common instance, and
- if r_1 and r_2 have a common instance, then $r_1 \simeq r_2$.

Note that if an rwo TRS$_+$ has a collapsing term rewriting rule, then all term rewriting rules of R are collapsing.

Example 2.1 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{\#\}$, $\Sigma_1 = \{f\}$, and $\Sigma_2 = \{g\}$. Consider the left-linear TRS$_+$s

- $R_1 = \{\, g(x_1, f(x_2)) \to f(\#),\ g(x_2, f(x_3)) \to x_1,\ g(x_1, \#) \to g(x_2, x_1) \,\}$,
- $R_2 = \{\, \# \to f(\#),\ g(x_1, \#) \to g(x_1, x_1),\ g(\#, x_1) \to g(x_1, x_1) \,\}$,
- $R_3 = \{\, f(x_1) \to x_1,\ f(f(x_2)) \to x_2 \,\}$,
- $R_4 = \{\, f(x_1) \to x_1,\ g(x_2, \#) \to x_2 \,\}$, and
- $R_5 = \{\, f(x_1) \to x_1,\ f(f(x_2)) \to g(x_2, x_2) \,\}$,
- $R_6 = \{\, f(x_1) \to f(x_2),\ g(f(x_2), x_3) \to g(x_2, x_2) \,\}$

over Σ. Then R_1 and R_4 are wa, and R_2, R_3, R_5, and R_6 are not wa. Furthermore, R_1, R_3, R_4, and R_5 are not rwo, and R_2, and R_6 are rwo. TRS$_+$s R_1, R_2, and R_6 are non-collapsing, and R_3, R_4, and R_5 are collapsing.

It is well known that we can decide for any two terms $s, t \in T_\Sigma(X)$, if they have a common instance [1]. Here we only recall the basic idea. We define $s' \in T_\Sigma(X)$ by renaming the variables of s such that $var(s') \cap var(t) = \emptyset$ holds, and then then using a unification algorithm we decide if s' and t are unifiable. If so, then s and t have a common instance, otherwise they do not. Consequently, we can decide whether a TRS$_+$ R is wa and whether R is rwo. To simplify the proofs, without loss of generality, we may reformulate the above concepts in the following way. In the definition of a wa TRS$_+$, we replace the relation $l_1 \simeq l_2$ by the equation $l_1 = l_2$. Moreover, we assume that for each left-hand side $l \in lhs(R)$, $var(l) = X_m$ for some $m \geq 0$. By this assumption, for each linear left-hand side $l \in lhs(R)$, we have $l \in CO_\Sigma(X_m)$ for some $m \geq 0$. In the definition of an rwo TRS$_+$ R, we replace the relation $r_1 \simeq r_2$ by the equation $r_1 = r_2$. Moreover, we assume that for each right-hand side $r \in rhs(R)$, $var(r) = X_m$ for some $m \geq 0$. By this assumption, for each linear right-hand side $r \in rhs(R)$, we have $r \in CO_\Sigma(X_m)$ for some $m \geq 0$. In this way, we get the following forms of the above definitions, which we will use from now on throughout the paper. We say that a TRS$_+$ R is wa if

- for any two left-hand sides $l_1, l_2 \in lhs(R)$,

- for all $\alpha \in (POS(l_1) - (\{\lambda\} \cup VPOS(l_1)))$, l_1/α and l_2 have no common instance, and
- if l_1 and l_2 have a common instance, then $l_1 = l_2$, and

- for each left-hand side $l \in lhs(R)$, $var(l) = X_m$ for some $m \geq 0$.

We say that a TRS_+ R is rwo if

- for any two right-hand sides $r_1, r_2 \in rhs(R)$,
  - for all $\alpha \in (POS(r_1) - (\{\lambda\} \cup VPOS(r_1)))$, r_1/α and r_2 have no common instance, and
  - if r_1 and r_2 have a common instance, then $r_1 = r_2$, and
- for each right-hand side $r \in rhs(R)$, $var(r) = X_m$ for some $m \geq 0$.

Example 2.2 We continue Example 2.1 We rename the variables in the term rewriting rules of the TRS_+s R_1, R_2, R_4, and R_6, so that they satisfy our assumptions and keep their properties. Let

- $R_1 = \{\, g(x_1, f(x_2)) \to f(\#),\ g(x_1, f(x_2)) \to x_3,\ g(x_1, \#) \to g(x_2, x_1) \,\}$,
- $R_2 = \{\, \# \to f(\#),\ g(x_1, \#) \to g(x_1, x_1),\ g(\#, x_1) \to g(x_1, x_1) \,\}$,
- $R_4 = \{\, f(x_1) \to x_1,\ g(x_1, \#) \to x_1 \,\}$, and
- $R_6 = \{\, f(x_2) \to f(x_1),\ g(f(x_1), x_2) \to g(x_1, x_1) \,\}$.

Then R_1 and R_4 are wa, and R_2 and R_6 are rwo.

We say that a TRS_+ R keeps the non-linear variables (R is knlv) if for any term rewriting rule $l \to r$ in R, each non-linear variable of l appears in r.

Example 2.3 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{\,\#\,\}$, $\Sigma_1 = \{\, f \,\}$, and $\Sigma_2 = \{\, g \,\}$. Consider the TRS_+s

$$R = \{\, g(x_1, f(x_1)) \to f(x_1),\ g(x_1, f(x_2)) \to \#,\ g(x_1, x_1) \to g(x_1, x_3) \,\}$$

and

$$S = \{\, \# \to f(x_1),\ g(x_1, x_1) \to \# \,\}$$

over Σ. Then R is knlv, and S is not knlv.

Let R be a TRS_+ over Σ. For any two terms $s, t \in T_\Sigma(X)$, position $\alpha \in POS(s)$, and term rewriting rule $l \to r$ in R with $l, r \in T_\Sigma(X_m)$, $m \geq 0$, we say that s rewrites to t applying the term rewriting rule $l \to r$ at α, and denote this by $s \to_{\alpha, l\to r} t$ if there are $s_1, \ldots, s_m \in T_\Sigma(X)$ such that $s/\alpha = l[s_1, \ldots, s_m]$ and $t = s[\alpha \leftarrow r[s_1, \ldots, s_m]]$. Here we also say that s rewrites to t and denote this by $s \to_R t$. A sequence of reductions

$$s_0 \to_{\alpha_1, l_1 \to r_1} s_1 \to_{\alpha_2, l_2 \to r_2} s_2 \to_{\alpha_3, l_3 \to r_3} \cdots \to_{\alpha_n, l_n \to r_n} s_n \text{ with } n \geq 0$$

is called a reduction sequence.

Dauchet and De Comité [4] introduced the inside-out and outside-in one-pass reductions with a TRS R in an intuitive way and illustrated these concepts by examples. Intuitively, for any two terms $s, t \in T_\Sigma(X)$, we say that s is rewritten to t in one-pass if we rewrite s into t applying some term rewriting rules such that the left-hand sides do not overlap. In case of an IO pass, we rewrite from the innermost of a bracketed expression of a term to the outermost, i.e., in a bottom-up order, hence the subtrees are rewritten before substituting and duplicating them. Furthermore in case of an OI pass, we rewrite from the outermost of a bracketed expression of a term to the innermost, i.e., in a top-down order, hence the subtrees are rewritten after substituting and duplicating them. We now formally define these concepts [23].

An inside-out (IO) one-pass reduction sequence with R is of the form

$$s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_n, l_n \to r_n}{\to} s_n, \tag{1}$$

where Conditions 1 and 2 hold:

1. $n \geq 0$, $s_0, \ldots, s_n \in T_\Sigma(X)$, and $\alpha_i \in POS(s_{i-1}) \cap POS(s_0)$ for $i = 1, \ldots, n$,

2. For any $2 \leq j \leq n$, $\{ \alpha_1, \ldots, \alpha_{j-1} \} \cap (\{ \gamma \in \mathbb{N}^* \mid \gamma \prec \alpha_j \} \cup \{ \alpha_j \xi \mid \xi \in (POS(l_j) - VPOS(l_j)) \}) = \emptyset$.

Informally, Condition 2 ensures that we rewrite in a bottom-up order and that the left-hand sides do not overlap. It says that for every $2 \leq j \leq n$, the positions $\alpha_1, \ldots, \alpha_{j-1}$ are no proper prefixes of α_j nor are positions of any nonvariable symbol in the occurrence of the left-hand side l_j of the term rewriting rule $l_j \to r_j$ when applying it at α_j in the jth step. Note that we can apply rules of which left-hand side is a variable several times at the same position.

The terms $s_0, \ldots, s_n$ are called IO one-pass sentential forms of s_0. For every term $s \in T_\Sigma(X)$, $IOSF(s)$ denotes the set of all IO one-pass sentential forms of s. For a tree language $L \subseteq T_\Sigma$, let $IOSF(L) = \bigcup_{s \in L} IOSF(s)$. We usually write (1) in the form

$$s_0 \underset{R,IO}{\to} s_1 \underset{R,IO}{\to} \cdots \underset{R,IO}{\to} s_n. \tag{2}$$

The notation $s_0 \Rightarrow_{R,IO} s_n$ means that there is an IO one-pass reduction sequence (1). When we want to refer to the integer n, we write $s_0 \Rightarrow_{R,IO,n} s_n$.

Example 2.4 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{ \# \}$, $\Sigma_1 = \{ f \}$, and $\Sigma_2 = \{ g \}$. Consider the left-linear TRS

$$R = \{ g(x_1, x_2) \to f(x_2),\ g(x_1, x_2) \to g(x_1, x_1) \}$$

over Σ. Then

$g(g(g(\#,\#),\#), g(\#,\#)) \to_{11, g(x_1,x_2) \to f(x_2)}$
$g(g(f(\#),\#), g(\#,\#)) \to_{1, g(x_1,x_2) \to f(x_2)} g(f(\#), g(\#,\#))$

is an IO one-pass reduction sequence with R. Furthermore,

$g(\#, g(f(\#),\#)) \to_{2, g(x_1,x_2) \to f(x_2)} g(\#, f(\#)) \to_{\lambda, g(x_1,x_2) \to g(x_1,x_1)} g(\#,\#)$

is another IO one-pass reduction sequence with R. However,

$g(\#, g(f(\#),\#)) \to_{\lambda, g(x_1,x_2) \to f(x_2)} f(g(f(\#),\#)) \to_{1, g(x_1,x_2) \to g(x_1,x_1)} f(g(f(\#), g(f(\#)))$ is not an IO one-pass reduction sequence with R.

Example 2.5 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{ \# \}$, $\Sigma_1 = \{ f \}$, and $\Sigma_2 = \{ g \}$. Consider the left-linear TRS$_+$

$$R = \{ x_1 \to f(x_2),\ x_1 \to \#,\ g(x_1, x_2) \to g(x_1, x_1) \}$$

over Σ. Then

$$g(g(g(\#, f(\#)),\#), g(\#,\#)) \to_{11, x_1 \to f(x_2)} g(g(f(\#),\#), g(\#,\#)) \to_{11, x_1 \to \#}$$

$$g(g(\#,\#), g(\#,\#)) \to_{2, x_1 \to \#} g(g(\#,\#),\#) \to_{1, g(x_1,x_2) \to g(x_1,x_1)} g(g(\#,\#), g(\#,\#))$$

is an IO one-pass reduction sequence with R. Furthermore,

$$g(\#, g(f(\#),\#)) \to_{2, g(x_1,x_2) \to g(x_1,x_1)} g(\#, g(f(\#), f(\#))) \to_{\lambda, g(x_1,x_2) \to g(x_1,x_1)}$$

$$g(\#,\#) \to_{\lambda, x_1 \to f(x_2)} f(f(\#))$$

is another IO one-pass reduction sequence with R. However,

$$g(\#, g(f(\#),\#)) \to_{\lambda, g(x_1,x_2) \to g(x_1,x_1)} g(\#,\#) \to_{2, x_1 \to f(\#)} g(\#, f(\#))$$

is not an IO one-pass reduction sequence with R.

An outside-in (OI) one-pass reduction sequence with R is of the form

$$s_0 \underset{\beta_1, l_1 \to r_1; \alpha_1}{\to} s_1 \underset{\beta_2, l_2 \to r_2; \alpha_2}{\to} s_2 \underset{\beta_3, l_3 \to r_3; \alpha_3}{\to} \cdots \underset{\beta_n, l_n \to r_n; \alpha_n}{\to} s_n, \tag{3}$$

where Conditions 1–5 below hold:

1. $n \geq 0$, $s_0, \ldots, s_n \in T_\Sigma(X)$, and $\alpha_i \in pos(s_0)$, $\beta_i \in pos(s_{i-1})$ for $i = 1, \ldots, n$.

2. $s_0 \rightarrow_{\beta_1, l_1 \to r_1} s_1 \rightarrow_{\beta_2, l_2 \to r_2} s_2 \rightarrow_{\beta_3, l_3 \to r_3} \cdots \rightarrow_{\beta_n, l_n \to r_n} s_n$ is a reduction sequence with R.

3. For all $1 \leq i < j \leq n$, β_j is not a proper prefix of β_i.

4. $\alpha_1 = \beta_1$.

5. For any $1 < j \leq n$, if there is $1 \leq i < j$ such that $\beta_i \preceq \beta_j$, then let k be the largest such i, and then

   (a) $\alpha_k \gamma \xi = \alpha_j$,

   (b) $\beta_k \delta \xi = \beta_j$, and

   (c) $lab(l_k, \gamma) = lab(r_k, \delta) \in X$

   for some $\gamma \in vpos(l_k)$, $\xi \in \mathbb{N}^*$, and $\delta \in vpos(r_k)$.

We note that the definition of an OI one-pass reduction sequence with R presented in [23] is incomplete, we corrected it by adding Item 3. It says that for all $1 \leq i < j \leq n$, β_i and β_j are incomparable or $\beta_i = \beta_j$. Informally, along (3), we rewrite in a top-down order, and keep on rewriting the unprocessed subtrees of the initial tree s_0. At the same time we keep track of the positions of the subtrees of the initial tree s_0. For every $i = 1, \ldots, n$, the position α_i points to the subtree s_0/α_i of s_0, and the position β_i points to an occurrence of s_0/α_i in s_{i-1}, to be rewritten in the ith step of (3) and of the reduction sequence appearing in Item 2. Item 3 says that along the reduction sequence in Item 2 we proceed in top-down manner. By the definition of k, Item 4 says that in the reduction sequence of Item 2 the rewrite step $s_{j-1} \rightarrow_{\beta_j, l_j \to r_j} s_j$ rewrites the input tree below the rewrite step $s_{k-1} \rightarrow_{\beta_k, l_k \to r_k} s_k$. We can view the jth step as R reduces the subtree $s_k/\delta_k\xi$, substituted for the variable $lab(r_k, \delta)$ in the right-hand side r_k in the kth step, at its position ξ. Formally, the terms $s_0, \ldots, s_n$ are called OI one-pass sentential forms of s_0. For every term $s \in T_\Sigma(X)$, $OISF(s)$ is the set of all OI one-pass sentential forms of s. That is, $OISF(s)$ is the set of all terms t such that there is an OI one-pass reduction sequence (3) with $s = s_0$ and $t = s_n$. For a tree language $L \subseteq T_\Sigma$, we put

$$OISF(L) = \bigcup_{s \in L} OISF(s).$$

We usually write (3) in the form

$$s_0 \underset{R,OI}{\rightarrow} s_1 \underset{R,OI}{\rightarrow} \cdots \underset{R,OI}{\rightarrow} s_n. \tag{4}$$

The notation

$$s_0 \underset{R,OI}{\Rightarrow} s_n$$

means that there is an OI one-pass reduction sequence (4). When we want to refer to the integer n, we write $s_0 \Rightarrow_{R,OI,n} s_n$.

Example 2.6 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{\#\}$, $\Sigma_1 = \{f\}$, and $\Sigma_2 = \{g\}$. Consider the left-linear TRS

$$R = \{ g(x_1, x_2) \to f(x_2),\ g(x_1, x_2) \to g(x_1, x_1) \}$$

over Σ. Then

$$\begin{aligned}
&g(g(g(\#,\#), g(\#,\#)), g(\#,\#)) \rightarrow_{\lambda, g(x_1,x_2) \to g(x_1,x_1); \lambda} \\
&g(g(g(\#,\#), g(\#,\#)), g(g(\#,\#), g(\#,\#))) \rightarrow_{1, g(x_1,x_2) \to f(x_2); 1} \\
&g(f(\#), g(g(\#,\#), g(\#,\#))) \rightarrow_{21, g(x_1,x_2) \to f(x_2); 11} \\
&g(f(\#), g(f(\#), g(\#,\#))) \rightarrow_{22, g(x_1,x_2) \to f(x_2); 12} g(f(\#), g(f(\#), f(\#))).
\end{aligned}$$

is an OI one-pass reduction sequence with R. Furthermore, we now list all OI one-pass sequences starting from $g(g(\#,\#),\#)$.

- $g(g(\#,\#),\#)$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow f(x_2);\lambda} f(\#)$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow f(x_2);1} g(f(\#), g(\#,\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow f(x_2);1}$ $g(f(\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow f(x_2);1} g(f(\#), f(\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow f(x_2);1}$ $g(f(\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow g(x_1,x_1);1} g(f(\#), g(\#,\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow g(x_1,x_1);1} g(g(\#,\#), g(\#,\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow g(x_1,x_1);1}$ $g(g(\#,\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow f(x_2);1} g(g(\#,\#), f(\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow g(x_1,x_1);1}$ $g(g(\#,\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow g(x_1,x_1);1} g(g(\#,\#), g(\#,\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow f(x_2);1} g(g(\#,\#), f(\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow f(x_2);1}$ $g(g(\#,\#), f(\#)) \rightarrow_{1, g(x_1,x_2)\rightarrow f(x_2);1} g(f(\#), f(\#))$,
- $g(g(\#,\#),\#) \rightarrow_{\lambda, g(x_1,x_2)\rightarrow g(x_1,x_1);\lambda} g(g(\#,\#), g(\#,\#)) \rightarrow_{2, g(x_1,x_2)\rightarrow f(x_2);1}$ $g(g(\#,\#), f(\#))) \rightarrow_{1, g(x_1,x_2)\rightarrow g(x_1,x_1);1} g(g(\#,\#), f(\#))$,
- $g(g(\#,\#),\#) \rightarrow_{1, g(x_1,x_2)\rightarrow f(x_2);1} g(f(\#),\#)$,
- $g(g(\#,\#),\#) \rightarrow_{1, g(x_1,x_2)\rightarrow g(x_1,x_1);1} g(g(\#,\#),\#)$.

Accordingly

$$OISF(g(g(\#,\#),\#)) = \{\, g(g(\#,\#),\#), f(\#), g(g(\#,\#), g(\#,\#)), g(f(\#), g(\#,\#)),$$
$$g(f(\#), f(\#)), g(g(\#,\#), f(\#)), g(f(\#),\#) \,\}.$$

For tree languages L and M over Σ, we say that L and M are IO one-pass joinable for R if $IOSF(L) \cap IOSF(M) \neq \emptyset$. For tree languages L and M over Σ, we say that L and M have a common IO one-pass ancestor with respect to R if there is a term $t \in T_\Sigma(X)$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$. We define the OI counterparts of the above definitions replacing IO by OI.

For the definition of the second-order IO (resp. OI) inclusion problem, the second-order IO (resp. OI) one-pass reachability problem, the second-order IO (resp. OI) one-pass joinability problem, and second-order IO (resp. OI) one-pass common ancestor problem, see the Introduction.

We now give a slightly weaker form of the second-order IO one-pass common ancestor problem and the second-order OI one-pass common ancestor problem requiring the common ancestor being a ground term.

Second-order IO one-pass common ground ancestor problem.
Instance: A TRS_+ R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is there a ground term $t \in T_\Sigma$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$?

Second-order OI one-pass common ground ancestor problem.
Instance: A TRS_+ R and recognizable tree languages L and M over a ranked alphabet Σ.
Question: Is there a ground term $t \in T_\Sigma$ such that $OISF(t) \cap L \neq \emptyset$ and $OISF(t) \cap M \neq \emptyset$?

Proposition 2.7 *For TRS$_+$s, (i) the second-order IO one-pass common ground ancestor problem and the second-order IO one-pass common ancestor problem are equivalent, and*

(ii) the second-order OI one-pass common ground ancestor problem and the second-order OI one-pass common ancestor problem are equivalent.

Proof. Let R be a TRS$_+$ and L and M be recognizable tree languages over a ranked alphabet Σ. If there are no constants in Σ, then $L = M = \emptyset$, and the answer is no to both the second-order IO one-pass common ancestor problem and the second-order OI one-pass common ancestor problem.

Assume that there is a constant $\$ \in \Sigma_0$, and let $t \in T_\Sigma(X)$ be such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$. We define s from t replacing all variables in t by \$. Then $IOSF(s) \cap L \neq \emptyset$ and $IOSF(s) \cap M \neq \emptyset$. Hence the answer to the second-order IO one-pass common ancestor problem is the same as the answer to the second-order IO one-pass common ground common ancestor problem. Analogously, the answer to the second-order OI one-pass common ancestor problem is the same as to the second-order OI one-pass common ground ancestor problem. $\square$

2.4 Recognizable Tree Languages

A bottom-up tree automaton (bta) over a ranked alphabet Σ is a quadruple $\mathcal{A} = (\Sigma, A, R, A_f)$, where A is a finite set of states of rank 0, $\Sigma \cap A = \emptyset$, $A_f (\subseteq A)$ is the set of final states, R is a finite set of term rewriting rules of the form

$$f(a_1, \ldots, a_m) \to a$$

with $m \geq 0$, $f \in \Sigma_m$, $a_1, \ldots, a_m, a \in A$. We call $f(a_1, \ldots, a_m)$ the left-hand side of the term rewriting rule $f(a_1, \ldots, a_m) \to a$. We consider R as a ground TRS$_+$ over the ranked alphabet $\Sigma \cup A$. For convenience, we write $\to_\mathcal{A}$ for $\to_R$. A state $a \in A$ is reachable if there is a tree $p \in T_\Sigma$ such that $p \to_\mathcal{A}^* a$. We say that $\mathcal{A}$ is connected if every state $a \in A$ is reachable. The bta $\mathcal{A} = (\Sigma, A, R, A_f)$ is total if for all $f \in \Sigma_m$, $m \geq 0$, and $a_1, \ldots, a_m \in A$, R has a term rewriting rule with the left-hand side $f(a_1, \ldots, a_m)$. The bta $\mathcal{A} = (\Sigma, A, R, A_f)$ is deterministic (dbta) if R has no two term rewriting rules with the same left-hand side. For any total dbta $\mathcal{A}$ and tree $t \in T_{\Sigma \cup A}$, there is exactly one state $a \in A$ such that $t \to_\mathcal{A}^* a$. We denote this a by $t^\mathcal{A}$. The tree language recognized by a bta $\mathcal{A}$ is $L(\mathcal{A}) = \{\, t \in T_\Sigma \mid \exists a \in A_f.\ t \to_\mathcal{A}^* a \,\}$. A tree language L is recognizable if there exists a bta $\mathcal{A}$ such that $L(\mathcal{A}) = L$ [8]. For every bta $\mathcal{A}$ we can construct a connected total dbta $\mathcal{B}$ such that $L(\mathcal{B}) = L(\mathcal{A})$. We give a recognizable tree language L via a connected total dbta $\mathcal{A}$ recognizing L.

For every total dbta $\mathcal{A}$, we can decide whether $L(\mathcal{A}) = \emptyset$. For all total dbtas $\mathcal{A}$ and $\mathcal{B}$ we can construct dbtas $\mathcal{C}$ and $\mathcal{D}$ such that $L(\mathcal{C}) = L(\mathcal{A}) \cap L(\mathcal{B})$ and $L(\mathcal{D}) = L(\mathcal{A}) - L(\mathcal{B})$. Consequently, for all total dbtas $\mathcal{A}$ and $\mathcal{B}$ we can decide whether $L(\mathcal{A}) \cap L(\mathcal{B}) \neq \emptyset$, and whether $L(\mathcal{A}) \subseteq L(\mathcal{B})$.

A generalized bottom-up tree automaton (gbta) over a ranked alphabet Σ is a quadruple $\mathcal{A} = (\Sigma, A, R, A_f)$, where A is a finite set of states of rank 0, $\Sigma \cap A = \emptyset$, $A_f (\subseteq A)$ is the set of final states, R is a finite set of term rewriting rules of the form

$$l[a_1, \ldots, a_m] \to a,$$

where $m \geq 0$, $l \in CO_\Sigma(X_m)$, and $a_1, \ldots, a_m, a \in A$ [7]. We call $l[a_1, \ldots, a_m]$ the left-hand side of the term rewriting rule $l[a_1, \ldots, a_m] \to a$. We consider R as a ground TRS$_+$ over the ranked alphabet $\Sigma \cup A$. For convenience, we write $\to_\mathcal{A}$ for $\to_R$. The tree language recognized by a gbta $\mathcal{A}$ is $L(\mathcal{A}) = \{\, t \in T_\Sigma \mid \exists a \in A_f.\ t \to_\mathcal{A}^* a \,\}$. For every gbta $\mathcal{A}$ we can construct a total dbta $\mathcal{B}$ such that $L(\mathcal{B}) = L(\mathcal{A})$ [7]. A state $a \in A$ is reachable if there is a tree $p \in T_\Sigma$ such that $p \to_\mathcal{A}^* a$, and $REACH_\mathcal{A}$ stands for the set of reachable states. We compute $REACH_\mathcal{A}$ as follows.

1. Let $i = 1$ and $REACH_1 = \{\, a \mid l \to a \in R_\mathcal{A}, l \in T_\Sigma \,\}$,
2. Let $i = i + 1$ and let W_i be the set of all states a such that
   - there is a term rewriting rule $l[a_1, \ldots, a_m] \to a \in R_\mathcal{A}$, with $l \in T_\Sigma(X_m)$, $m \geq 1$ and
   - for every $1 \leq i \leq m$, if the variable x_i appears in l, then $a_i \in REACH_{i-1}$.

Let $REACH_i = REACH_{i-1} \cup W_i$. If $REACH_i = REACH_{i-1}$, then let $REACH = REACH_i$, otherwise goto 1.

It is not hard to see that our algorithm terminates with $REACH = REACH_{\mathcal{A}}$.

Let

$$z_0 \to_{l_1 \to r_1, \alpha_1} z_1 \to_{l_2 \to r_2, \alpha_2} z_2 \to_{l_3 \to r_3, \alpha_3} \cdots \to_{l_k \to r_k, \alpha_k} z_k \text{ with } k \geq 0 \qquad (D)$$

be a reduction sequence, and let $\beta \in POS(z_0)$. Then we define the restriction

$$v_0 \to_{\mathcal{A}} v_1 \to_{\mathcal{A}} \cdots \to_{\mathcal{A}} v_\ell \text{ with } 0 \leq \ell \leq k \qquad (D_\beta)$$

of (D) to β as follows.

Let $v_0 = z_0$, $m = 0$ and $j = 0$. If $k = 0$, then we have (D_β) with $\ell = 0$. Otherwise, goto 1.

1. Let $m = m + 1$. If β is a prefix of α_m, then let $j = j + 1$, and we define $\gamma_j \in N^*$ such that $\alpha_m = \beta\gamma_j$ holds. Moreover, $\mathcal{A}$ applies the term rewriting rule $l_m \to r_m$ at γ_j in D_β. That is, let v_j be such that $v_{j-1} \to_{l_m \to r_m} v_j$. If $m = k$, then we have (D_β) with $\ell = j$. Otherwise, goto 1.

Statement 2.8 *Let $\mathcal{A} = (\Sigma, A, R, A_f)$ be a gbta, and let*

$$s_0 \to_{\mathcal{A}} s_1 \to_{\mathcal{A}} \cdots \to_{\mathcal{A}} s_k \text{ with } k \geq 0 \qquad (D)$$

be a reduction sequence where $s_0 \in T_\Sigma$ and $s_i \in T_{\Sigma \cup A}$ for $i = 1, \ldots, k$. Then there exist $u \in CO_\Sigma(X_m)$ and $t_1, \ldots, t_m \in T_\Sigma$ with $m \geq 0$ such that

- *$s_0 = u[t_1, \ldots, t_m]$,*
- *$s_k = u[a_1, \ldots, a_m]$ for some $a_1, \ldots, a_m \in A$, and*
- *for each $i = 1, \ldots, m$, the restriction of (D) to $adr(u, x_i)$ is of the form*

  $$t_i = z_{i0} \to_{\mathcal{A}} z_{i1} \to_{\mathcal{A}} \cdots \to_{\mathcal{A}} z_{ik_i} = a_i \qquad (D_{adr(u,x_i)})$$

  for some $k_i \geq 0$ and $z_{in} \in T_{\Sigma \cup A}$ for $n = 1, \ldots, k_i$.

Proof. By induction on the length k of the reduction sequence (D). $\square$

3 IO One-Pass Reductions

We show that for any right-linear knlv rwo TRS_+ R and recognizable tree language L, $IOSF(L)$ is recognizable. For knlv rwo TRS_+s the second-order IO one-pass inclusion problem and the second-order IO one-pass reachability problem are decidable. Furthermore, for right-linear knlv rwo TRS_+s the second-order IO one-pass joinability problem is decidable. For left-linear TRS_+s, the second-order IO one-pass common ancestor problem is decidable.

The following example shows that for any left-linear knlv rwo TRS_+ R and recognizable tree language L, $IOSF(L)$ is not necessarily recognizable.

Example 3.1 Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{\#\}$, $\Sigma_1 = \{f\}$, and $\Sigma_2 = \{g\}$. Consider the left-linear knlv rwo TRS

$$R = \{ f(x_1) \to g(x_1, x_1) \}$$

over Σ. Note that R is a wa TRS as well. Let $f^0(\#) = \#$, and $f^{n+1}(\#) = f(f^n(\#))$ for every $n \geq 0$. Then let $L = \{ f^n(\#) \mid n \geq 0 \}$. When applying the term rewriting rule $f(x_1) \to g(x_1, x_1)$ along an IO reduction of a tree $s \in L$, R duplicates a proper subtree t of s. Hence for any tree $t \in IOSF(f^n(\#))$ and position $\alpha \in POS(t)$ with $lab(t, \alpha) = \#$, the length of α is n, that is, in t the branches from root to leaf are of length n. If $IOSF(L)$ were recognizable, then the branches of the resulting forks could be pumped as in the proof of the pumping lemma for string languages or tree languages. This yields trees in $IOSF(L)$ where the branches from root to leaf are of different length. This contradicts our observation on the elements of the set $IOSF(L)$. Hence $IOSF(L)$ is not a recognizable tree language. One can show similarly, that $OISF(L)$ is not recognizable either.

In the light of Example 3.1, for a recognizable tree language L, we study the set $IOSF(L)$ for right-linear knlv rwo TRS_+s.

Theorem 3.2 *For any right-linear knlv rwo TRS_+ R and recognizable tree language L over a ranked alphabet Σ, we can construct a gbta $\mathcal{C}$ such that $L(\mathcal{C}) = IOSF(L)$.*

We now outline the proof of the theorem. Let R be a right-linear knlv rwo TRS_+ over a ranked alphabet Σ, L be a recognizable tree language over Σ, and let the connected total dbta $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be such that $L(\mathcal{A}) = L$. We construct a gbta $\mathcal{C}$ over Σ, and illustrate it by examples. Then we show Lemmas 3.6, 3.8, 3.9, and 3.10, which describe the connections between the IO one-pass reduction sequences of R, the dbta $\mathcal{A}$, and the gbta $\mathcal{C}$. Finally we show Lemma 3.11 and Lemma 3.12, which imply $L(\mathcal{C}) = IOSF(L)$.

An input t of $\mathcal{C}$ is an output of an IO one-pass reduction sequence of $\mathcal{R}$ starting from some $p \in L$, and $\mathcal{C}$ processes t mimicking $\mathcal{A}$ on all trees $\overline{p}$ such that $\overline{p} \Rightarrow_{R,IO} t$. At each node of t, $\mathcal{C}$ has a choice

- to consider this node as a node of the input term p as well, and to mimic $\mathcal{A}$ on it, or
- to attempt to recognize the right-hand side r of a term rewriting rule $l \to r$, and in case of success to simultaneously mimic $\mathcal{A}$ on all left-hand sides l such that $l \to r \in R$.

In this way, $\mathcal{C}$ mimics $\mathcal{A}$ on all input trees $\overline{p}$ with $\overline{p} \Rightarrow_{R,IO} t$ resulting in the set of the states $\overline{p}^{\mathcal{A}}$. Thus the states of $\mathcal{C}$ are the state sets in $P_+(A)$ rather than the states of $\mathcal{A}$, and when $\mathcal{C}$ reduces t to a state, $\mathcal{C}$ collects together the states $p^{\mathcal{A}}$ in its state. That is, gbta $\mathcal{C}$ computes for any tree $t \in T_\Sigma$, the state set $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t \}$. We explain the computation of $\mathcal{C}$ in the following intuitive way. Along an IO one-pass reduction sequence, several input subtrees p may be reduced by R to the same output tree t. In this way, along an IO one-pass reduction sequence an instance of a non-linear left-hand side may appear as a result of rewriting different input subtrees to the same tree. That is, a subtree $u[p_1, \ldots, p_k]$ of the input tree is not an instance of any left-hand side in $lhs(R)$, however, rewriting p_i into q_i for $i = 1, \ldots, k$, R rewrites $u[p_1, \ldots, p_k]$ into $u[q_1, \ldots, q_k] = l[t_1, \ldots, t_n]$, which is an instance of some left-hand side $l \in lhs(R)$. We illustrate this phenomenon by the following example.

Example 3.3 Let $\Sigma = \Sigma_0 \cup \Sigma_2$, where $\Sigma_0 = \{ \#, \$ \}$, $\Sigma_2 = \{ f \}$. Consider the TRS

$$R = \{ f(x_1, x_1) \to \$ \}.$$

Then the tree $f(f(\$, \$), \$)$ is not an instance of any left-hand side in $lhs(R)$. However, we have $f(f(\$, \$), \$) \to_{R,IO} f(\$, \$)$, and $f(\$, \$)$ is an instance of the left-hand side $f(x_1, x_1) \in lhs(R)$.

When gbta $\mathcal{C}$ recognizes a right-hand side $r \in rhs(R)$ with $r \in CO_\Sigma(X_m)$ and $m \geq 0$, $\mathcal{C}$ simultaneously mimic $\mathcal{A}$ on all left-hand sides l such that $l \to r \in R$ in the following way. Assume that R substitutes in r the output subtree t_i for the non-linear variable x_i of l for some $i \in \{ 1, \ldots, m \}$ and along its computation $\mathcal{C}$ arrives in state $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t_i \}$ at x_i. Then $\mathcal{C}$ substitutes the elements of $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t_i \}$ for all occurrences of the non-linear variable x_i in all left-hand sides l with $l \to r \in R$. In this way, each non-linear variable x_i in any left-hand side l with $l \to r \in R$ gets substituted, because x_i also appears in the right-hand side r, as R is knlv. Our assumption that R is rwo ensures that when $\mathcal{C}$ recognizes a right-hand side $r \in rhs(R)$, it does not ignore another overlapping right-hand side. Thus $\mathcal{C}$ can compute the set $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t \}$ for each input tree t.

For any knlv rwo TRS_+ R and connected total dbta $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$, we now introduce the gbta $\mathcal{C}(R, \mathcal{A})$, denoted simply by $\mathcal{C}$ when R and $\mathcal{A}$ are understood from the context. Let $\mathcal{C} = (\Sigma, P_+(A), R_{\mathcal{C}}, C_f)$, where $C_f = \{ W \mid W \in P_+(A) \text{ and } W \cap A_f \neq \emptyset \}$ and $R_{\mathcal{C}}$ consists of the following term rewriting rules of two types:

Type 1. For all $f \in \Sigma_m$ with $m \geq 0$, all $W_1, \ldots, W_m, W \in P_+(A)$ such that

$$W = \{ a \in A \mid f(a_1, \ldots, a_m) \to a \in R_{\mathcal{A}} \text{ and } a_1 \in W_1, \ldots, a_m \in W_m \}$$

let $R_{\mathcal{C}}$ contain the term rewriting rule $f(W_1, \ldots, W_m) \to W$.

Type 2. For every right-hand side $r \in rhs(R)$ with assuming that $r \in T_\Sigma(X_m)$, $var(r) = X_m$ for some $m \geq 0$, and for all $W_1, \ldots, W_m \in P_+(A)$, let $R_\mathcal{C}$ contain the term rewriting rule

$$r[W_1, \ldots, W_m] \to W, \tag{5}$$

where W consists of all states $a \in A$ such that

(1) $r[a_1, \ldots, a_m] \to_\mathcal{A}^* a$ for some $a_1 \in W_1, ..., a_m \in W_m$, or

(2) there is a term rewriting rule $l \to r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

- $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{ x_{i_1}, \ldots, x_{i_k} \} = var(l)$ and
- $l'[a_1, \ldots, a_k] \to_\mathcal{A}^* a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in W_{i_j}$, otherwise $a_j \in A$.

Then we say that the right-hand side $r \in rhs(R)$ yields the term rewriting rule $r[W_1, \ldots, W_m] \to W$.

Note that if r is a variable, that is $r = x_i$ for some $i \geq 1$, then the type 2 term rewriting rule $r[W_1, \ldots, W_m] \to W \in R_\mathcal{C}$ is of the form $W_i \to W$. By the definition of type 2 term rewriting rules, for each $V \in P_+(A)$, there is a unique term rewriting rule $V \to W$ in $R_\mathcal{C}$ with left-hand side V. Furthermore, for each term rewriting rule $V \to W \in R_\mathcal{C}$, we have $V \subseteq W$. Hence for each $V \in P_+(A)$, there is a unique set $Z \in P_+(A)$ such that $V \to_\mathcal{C}^* Z$ and $Z \to Z \in R_\mathcal{C}$. We call Z the $\mathcal{C}$ closure of V and denote Z by $Cl_\mathcal{C}(V)$.

We now continue our intuitive explanation. Assume that $\mathcal{C}$ recognizes a right-hand side $r \in rhs(R)$. Then $\mathcal{C}$ computes a state $W \in P_+(A)$ on r. To compute W, $\mathcal{C}$ mimics $\mathcal{A}$ on all left-hand sides l with $l \to r \in R$ in the following way. If a variable x_i does not appear in the right-hand side r, then we substitute an arbitrary state $a \in A$ of $\mathcal{A}$ for x_i. On the other hand, to each variable x_i of the right-hand side r, we assign a state W_i, where $\mathcal{C}$ arrives to the occurrence of x_i in r in state W_i. Then for each left-hand side l with $l \to r \in R$, for each variable x_i of l, for each ocurrence of x_i we substitute an arbitrary state $a_i \in W_i$ of $\mathcal{A}$. If x_i is a non-linear variable of l, then it also appears in the right-hand side r because R is a knlv TRS. In this way, the value of x_i becomes a state $a_i \in W_i$ rather than an arbitrary state $a \in A$. Note that when gbta $\mathcal{C}$ recognizes the right-hand side $r \in rhs(R)$ of a term rewriting rule $l \to r$ and mimic $\mathcal{A}$ on the left-hand side l, $\mathcal{C}$ may substitute different states of W_i for the occurrences of a non-linear variable x_i in l. This is because R may have reduced different input subtrees – evaluated by $\mathcal{A}$ to different states in W_i – to the same tree before applying the term rewriting rule $l \to r \in R$, and the right-hand side r – appearing in the resulting output tree – is then evaluated by $\mathcal{C}$. To ensure this in Condition (2) we introduce the linear tree $l' \in CO_\Sigma(X_k)$, $k \geq 0$, such that l is an instance of l', i.e., $l = l'[x_{i_1}, \ldots, x_{i_k}]$. Here k is equal to the number of occurrences of variables in l, and the variables $x_{i_1}, \ldots, x_{i_k}$ are not necessarily different, and $var(l) = \{ x_{i_1}, \ldots, x_{i_k} \} = X_h$ for some $h \geq 0$. In this way, we describe the possible scenario that there are $p_1, \ldots, p_k \in T_\Sigma$, $q_1, \ldots, q_k \in T_\Sigma$, and $s_1, \ldots, s_h \in T_\Sigma$ such that R reduces an input subtree $l'[p_1, \ldots, p_k]$ to $l'[q_1, \ldots, q_k]$, where $q_j = s_{i_j}$ for $j = 1, \ldots, k$, and hence $l'[q_1, \ldots, q_k] = l'[s_{i_1}, \ldots, s_{i_k}] = l[s_1, \ldots, s_h]$. Then we put $l'[a_1, \ldots, a_k]^\mathcal{A}$ in W where for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in W_{i_j}$, otherwise $a_j \in A$. In this fashion we achieved that non-linear variables may appear in the left-hand sides of the term rewriting rules of R. Since R is an rwo TRS, if $\mathcal{C}$ recognizes a right-hand side $r \in rhs(R)$, then during recognizing r, $\mathcal{C}$ did not pass by a right-hand side $r' \in rhs(R)$, and hence $\mathcal{C}$ did not miss out the computation of $\mathcal{A}$ on a left-hand side l' with $l' \to r' \in R$. In this way, on its input subtree t, $\mathcal{C}$ mimics $\mathcal{A}$ simultaneously on all input subtrees p of $\mathcal{A}$ with $p \Rightarrow_{R,IO} t$ resulting in the set of the states $p^\mathcal{A}$. When $\mathcal{C}$ reduces t to a state $W \in P_+(A)$, $\mathcal{C}$ collects together all the states $p^\mathcal{A}$ in W.

Our assumption that R is not necessarily left-linear requires that R is a knlv rwso TRS, and makes the construction of $\mathcal{C}$ sophisticated and the proof of the theorem original and hard. We now present two examples for the construction of the gbta $\mathcal{C}$.

Example 3.4 Let $\Sigma = \Sigma_0 \cup \Sigma_2$, where $\Sigma_0 = \{ \#, \$ \}$ and $\Sigma_2 = \{ f \}$. Let

$$L = \{ t \in T_\Sigma \mid \# \text{ appears an even number times in } t \}.$$

Consider the dbta $\mathcal{A} = (\Sigma, \{0,1\}, R_{\mathcal{A}}, \{0\})$, where $R_{\mathcal{A}}$ consists of the following term rewriting rules:

$$\# \to 1,\ \$ \to 0,$$

$$f(0,0) \to 0,\ f(0,1) \to 1, f(1,0) \to 1, f(1,1) \to 0.$$

Then for every $t \in T_\Sigma$, $t^{\mathcal{A}} = 0$ if and only if $\#$ appears an even number times in t. Hence $L = L(\mathcal{A})$.

Consider the knlv rwo TRS$_+$

$$R = \{\, f(x_1, x_2) \to \#,\ f(f(x_1, x_1), x_3) \to f(f(x_1, x_1), x_2)\ \}$$

over Σ. Then $\mathcal{C} = (\Sigma, P_+(\{0,1\}), R_{\mathcal{C}}, \{\{0\}, \{0,1\}\})$, where $R_{\mathcal{C}}$ consists of the following term rewriting rules of two types.

Type 1 term rewriting rules:

$$\# \to \{1\},\ \$ \to \{0\},$$

$$f(\{0\},\{0\}) \to \{0\},\ f(\{0\},\{1\}) \to \{1\},\ f(\{0\},\{0,1\}) \to \{0,1\},$$

$$f(\{1\},\{0\}) \to \{1\},\ f(\{1\},\{1\}) \to \{0\},\ f(\{1\},\{0,1\}) \to \{0,1\},$$

$$f(\{0,1\},\{0\}) \to \{0,1\},\ f(\{0,1\},\{1\}) \to \{0,1\},\ f(\{0,1\},\{0,1\}) \to \{0,1\}.$$

Type 2 term rewriting rules:

$$\# \to \{0,1\},$$

$$f(f(\{0\},\{0\}),\{0\}) \to \{0,1\},\ f(f(\{0\},\{0\}),\{1\}) \to \{0,1\},$$

$$f(f(\{0\},\{0\}),\{0,1\}) \to \{0,1\},$$

$$f(f(\{1\},\{1\}),\{0\}) \to \{0,1\},\ f(f(\{1\},\{1\}),\{1\}) \to \{0,1\},$$

$$f(f(\{1\},\{1\}),\{0,1\}) \to \{0,1\},$$

$$f(f(\{0,1\},\{0,1\}),\{0\}) \to \{0,1\},\ f(f(\{0,1\},\{0,1\}),\{1\}) \to \{0,1\},$$

$$f(f(\{0,1\},\{0,1\}),\{0,1\}) \to \{0,1\}.$$

Example 3.5 Let Σ, L and $\mathcal{A}$ be the same as in Example 3.4. Consider the right-linear knlv rwo TRS$_+$

$$R = \{\, f(f(x_1, x_1), x_2) \to f(f(x_1, x_2), \$)\ , f(f(x_1, x_2), f(x_1, x_2)) \to f(f(x_1, x_2), \$)\,\}$$

over Σ. We have the IO one-pass reductions:

$$f(f(\#,\#),\#) \to_{R,IO} f(f(\#,\#),\$)$$

$$f(f(\#,\#), f(\#,\#)) \to_{R,IO} f(f(\#,\#),\$)$$

Observe that R reduces two different trees, $f(f(\#,\#),\#)$ and $f(f(\#,\#), f(\#,\#))$ to the same tree, $f(f(\#,\#),\$)$. Hence along the IO reduction

$$f(f(f(\#,\#),\#), f(f(\#,\#), f(\#,\#))) \to_{R,IO} f(f(f(\#,\#),\$), f(f(\#,\#), f(\#,\#))) \to_{R,IO}$$

$$f(f(f(\#,\#),\$), f(f(\#,\#),\$))$$

R reduces two different input subtrees, $f(f(\#,\#),\#)$ and $f(f(\#,\#), f(\#,\#))$ to the same tree, $f(f(\#,\#),\$)$.

Then $\mathcal{C} = (\Sigma, P_+(\{0,1\}), R_{\mathcal{C}}, \{\{0\}, \{0,1\}\})$, where $R_{\mathcal{C}}$ consists of the following term rewriting rules of two types.

Type 1 term rewriting rules:

$$\# \to \{1\},\ \$ \to \{0\},$$

$$f(\{0\},\{0\}) \to \{0\},\ f(\{0\},\{1\}) \to \{1\},\ f(\{0\},\{0,1\}) \to \{0,1\},$$

$$f(\{1\},\{0\}) \to \{1\},\ f(\{1\},\{1\}) \to \{0\},\ f(\{1\},\{0,1\}) \to \{0,1\},$$

$$f(\{0,1\},\{0\}) \to \{0,1\},\ f(\{0,1\},\{1\}) \to \{0,1\},\ f(\{0,1\},\{0,1\}) \to \{0,1\}.$$

Type 2 term rewriting rules:

The term rewriting rule $f(f(x_1, x_1), x_2) \to f(f(x_1, x_2), \$)$ of R produces the term rewriting rules

$$f(f(\{0\},\{0\}),\$) \to \{0\},\ f(f(\{0\},\{1\}),\$) \to \{1\},\ f(f(\{0\},\{0,1\}),\$) \to \{0,1\},$$

$$f(f(\{1\},\{0\}),\$) \to \{0\},\ f(f(\{1\},\{1\}),\$) \to \{1\},\ f(f(\{1\},\{0,1\}),\$) \to \{0,1\},$$

$$f(f(\{0,1\},\{0\}),\$) \to \{0\},\ f(f(\{0,1\},\{1\}),\$) \to \{1\},\ f(f(\{0,1\},\{0,1\}),\$) \to \{0,1\}$$

of $\mathcal{C}$.

The term rewriting rule $f(f(x_1,x_2),f(x_1,x_2)) \to f(f(x_1,x_2),\$)$ produces the term rewriting rules

$$f(f(\{0\},\{0\}),\$) \to \{0\},\ f(f(\{0\},\{1\}),\$) \to \{0\},\ f(f(\{0\},\{0,1\}),\$) \to \{0\},$$

$$f(f(\{1\},\{0\}),\$) \to \{0\},\ f(f(\{1\},\{1\}),\$) \to \{0\},\ f(f(\{1\},\{0,1\}),\$) \to \{0\},$$

$$f(f(\{0,1\},\{0\}),\$) \to \{0\},\ f(f(\{0,1\},\{1\}),\$) \to \{0\},\ f(f(\{0,1\},\{0,1\}),\$) \to \{0\}$$

of $\mathcal{C}$.

Lemma 3.6 *Let R be a right-linear knlv rwo TRS_+ over a ranked alphabet Σ and $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$ be a connected total dbta. For every $t \in T_\Sigma$ and $V \in P_+(A)$, if a reduction sequence*

$$t = s_0 \underset{\mathcal{C}}{\to} s_1 \underset{\mathcal{C}}{\to} \cdots \underset{\mathcal{C}}{\to} s_n = V \text{ with } n \geq 0 \tag{6}$$

holds, then for every $a_0 \in V$, there is $p \in T_\Sigma$ such that $p^\mathcal{A} = a_0$ and $p \Rightarrow_{R,IO} t$.

Proof. We proceed by induction on the number ν of applications of type 2 term rewriting rules along the reduction sequence (6).

Base of induction: $\nu = 0$. By the definition of type 1 term rewriting rules of $R_\mathcal{C}$, by induction on n we can show that $V = \{t^\mathcal{A}\}$. Let $p = t$. Obviously, $p^\mathcal{A} = t^\mathcal{A}$ and $p \Rightarrow_{R,IO} t$.

Induction step: Let $\nu \geq 0$, and assume that for all integers less than or equal to ν, the statement holds. We now show that the statement holds for $\nu + 1$ as well. Let $a_0 \in V$ be arbitrary. Let $0 \leq j \leq n-1$ be such that

- in the $j+1$th step $s_j \to_\mathcal{C} s_{j+1}$ of (6), $\mathcal{C}$ applies a type 2 term rewriting rule $r[W_1,\ldots,W_m] \to W \in R_\mathcal{C}$, yielded by some right-hand side $r \in rhs(R)$, where $r \in CO_\Sigma(X_m)$, $m \geq 0$, $W_1,\ldots,W_m, W \in P_+(A)$, and
- along the last $n-j-1$ steps $s_{j+1} \to_\mathcal{C} s_{j+2} \to_\mathcal{C} \cdots \to_\mathcal{C} s_n$ of (6), $\mathcal{C}$ applies only term rewriting rules of type 1.

For each $i = 1,\ldots,m$, let $\alpha_i \in POS(r)$ be such that $lab(r,\alpha_i) = x_i$. Then by Statement 2.8, there exist $u \in CO_\Sigma(X_{\ell+1})$ and $t_1,\ldots,t_m, v_1,\ldots,v_\ell \in T_\Sigma$ with $m,\ell \geq 0$ such that

(a) $t = u[r[t_1,\ldots,t_m], v_1,\ldots,v_\ell]$,

(b) $s_j = u[r[W_1,\ldots,W_m], V_1,\ldots,V_\ell]$ for some $W_1,\ldots,W_m \in P_+(A)$, $V_1,\ldots,V_\ell \in P_+(A)$,

(c) for each $i = 1,\ldots,m$, the restriction of (6) to $adr(u,x_1)\alpha_i$ is of the form

$$t_i = z_{i0} \to_\mathcal{C} z_{i1} \to_\mathcal{C} \cdots \to_\mathcal{C} z_{ik_i} = W_i$$

for some $k_i \geq 0$ and $z_{i1},\ldots,z_{ik_i} \in T_{\Sigma \cup P_+(A)}$,

(d) for each $i = 1,\ldots,\ell$, the restriction of (6) to $adr(u,x_{i+1})$ is of the form

$$v_i = w_{i0} \to_\mathcal{C} w_{i1} \to_\mathcal{C} \cdots \to_\mathcal{C} w_{i\vartheta_i} = V_i$$

for some $\vartheta_i \geq 0$ and $w_{i1},\ldots,w_{i\vartheta_i} \in T_{\Sigma \cup P_+(A)}$, and

(e) $s_{j+1} = u[W, V_1,\ldots,V_\ell]$.

By the definition of type 1 term rewriting rules of $R_{\mathcal{C}}$, by induction on $n-j-1$, we can show that

$$V = \{\, a' \mid u[a, c_1, \ldots, c_\ell] \xrightarrow[\mathcal{A}]{*} a' \text{ for some } a \in W, c_1 \in V_1, \ldots, c_\ell \in V_\ell \,\}.$$

Hence

$u[a, c_1, \ldots, c_\ell] \to^*_{\mathcal{A}} a_0$ for some $a \in W$ and $c_1 \in V_1, \ldots, c_\ell \in V_\ell$.

Recall that $r \in rhs(R)$, with assuming that $r \in CO_\Sigma(X_m)$ for some $m \geq 0$, yields the type 2 term rewriting rule $r[W_1, \ldots, W_m] \to W \in R_{\mathcal{C}}$. By the definition of type 2 term rewriting rules of $R_{\mathcal{C}}$, W consists of all states $a \in A$ such that Condition (1) or Condition (2) holds.

First, assume that Condition (1) in the definition of type 2 term rewriting rules holds for a. That is to say, $r[a_1, \ldots, a_m] \to^*_{\mathcal{A}} a$ for some $a_1 \in W_1, ..., a_m \in W_m$. Then by the induction hypothesis,

for every $i = 1, \ldots, m$, there is $p_i \in T_\Sigma$ such that $p_i^{\mathcal{A}} = a_i$ and $p_i \Rightarrow_{R,IO} t_i$.

Again by the induction hypothesis,

for every $j = 1, \ldots, \ell$, there is $q_j \in T_\Sigma$ such that $q_j^{\mathcal{A}} = c_j$ and $q_j \Rightarrow_{R,IO} v_j$.

Then let $p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell]$. Therefore

- $p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell] \to^*_{\mathcal{A}} u[r[a_1, \ldots, a_m], c_1, \ldots, c_\ell] \to^*_{\mathcal{A}} u[a, c_1, \ldots, c_\ell] \to^*_{\mathcal{A}} a_0$, and
- there is an IO one-pass reduction sequence

$$p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell] = z_0 \to_R z_1 \to_R \cdots \to_R u[r[t_1, \ldots, t_m], v_1, \ldots, v_\ell] = t.$$

Second, assume that Condition (2) in the definition of type 2 term rewriting rules holds for a. That is, there is a term rewriting rule $l \to r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

- $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{\, x_{i_1}, \ldots, x_{i_k} \,\} = var(l)$ and
- $l'[a_1, \ldots, a_k] \to^*_{\mathcal{A}} a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in W_{i_j}$, otherwise $a_j \in A$.

For every $j = 1, \ldots, k$, we distinguish two cases.

Case 1: $x_{i_j} \in var(r)$. Then $a_j \in W_{i_j}$. Hence by the induction hypothesis, there is $p_j \in T_\Sigma$ such that $p_j^{\mathcal{A}} = a_j$ and $p_j \Rightarrow_{R,IO} t_{i_j}$.

Case 2: $x_{i_j} \notin var(r)$. Then let $p_j \in T_\Sigma$ be arbitrary.

Since R is knlv, if x_{i_j} is a non-linear variable of l, then $x_{i_j} \in var(r)$. Accordingly

$$l'[p_1, \ldots, p_k] \underset{R,IO}{\Rightarrow} l'[t_{i_1}, \ldots, t_{i_k}] = l[t_1, \ldots, t_h].$$

Again by the induction hypothesis,

for every $j = 1, \ldots, \ell$, there is $q_j \in T_\Sigma$ such that $q_j^{\mathcal{A}} = c_j$ and $q_j \Rightarrow_{R,IO} v_j$.

Then let

$$p = u[l'[p_1, \ldots, p_k], q_1, \ldots, q_\ell].$$

Therefore

- $p = u[l'[p_1, \ldots, p_k], q_1, \ldots, q_\ell] \to^*_{\mathcal{A}} u[l'[a_1, \ldots, a_k], c_1, \ldots, c_\ell] \to^*_{\mathcal{A}} u[a, c_1, \ldots, c_\ell] \to^*_{\mathcal{A}} a_0$, and
- there is an IO one-pass reduction sequence

$$p = u[l'[p_1, \ldots, p_k], q_1, \ldots, q_\ell] = s_0 \underset{R}{\to} s_1 \underset{R}{\to} \cdots \underset{R}{\to} u[l'[t_{i_1}, \ldots, t_{i_k}], v_1, \ldots, v_\ell] =$$

$$u[l[t_1, \ldots, t_h], v_1, \ldots, v_\ell] \underset{R}{\to} u[r[t_1, \ldots, t_m], v_1, \ldots, v_\ell] = t.$$

□

Note that in the last step of the above proof we used our assumption that R is right-linear and hence $r \in rhs(R)$ is linear as well. If r is not linear, then Condition (a) does not necessarily hold, instead of (a) we have

(a1) $r = r'[x_{j_1}, \ldots, x_{j_\mu}]$ for some $r' \in CO_\Sigma(X_\mu)$, $\mu \geq 0$, and $\{ x_{j_1}, \ldots, x_{j_\mu} \} = var(r)$, and

(a2) $t = u[r'[t_1, \ldots, t_\mu], v_1, \ldots, v_\ell]$ for some $u \in T_\Sigma(X_{\ell+1})$, $\ell \geq 0$, $t_1, \ldots, t_{j_\mu} \in T_\Sigma$ and $v_1, \ldots, v_\ell \in T_\Sigma$.

It may be the case that there are $\chi, \xi \in \{ 1, \ldots, \mu \}$ such that $x_{j_\chi} = x_{j_\xi}$ however $t_\chi \neq t_\xi$. Hence the last step

$$u[l[t_1, \ldots, t_h], v_1, \ldots, v_\ell] \underset{R}{\rightarrow} u[r'[t_1, \ldots, t_\mu], v_1, \ldots, v_\ell] = t$$

of the above proof does not hold.

Definition 3.7 Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$ be a connected total dbta. For every $t \in T_\Sigma$, we define the state sets $Q\langle t, R, \mathcal{A}\rangle \in P_+(A)$ and $U\langle t, R, \mathcal{A}\rangle \in P_+(A)$ – denoted simply as $Q\langle t\rangle$ and $U\langle t\rangle$, respectively, when R and $\mathcal{A}$ are understood from the context – by induction on $height(t)$.

Base of induction: $height(t) = 0$. Then $t = f$ for some $f \in \Sigma_0$. First we define $Q\langle t\rangle \in P_+(A)$ and then $U\langle t\rangle \in P_+(A)$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then let

$$Q\langle t\rangle = \{ a \in A \mid f \rightarrow a \in R_\mathcal{A} \}.$$

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $t = r$. Let $Q\langle t\rangle$ consist of all states $a \in A$ such that

(1) $f \rightarrow a \in R_\mathcal{A}$ or

(2) there is a term rewriting rule $l \rightarrow r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

- $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{ x_{i_1}, \ldots, x_{i_k} \} = var(l)$, and
- $l'[a_1, \ldots, a_k] \rightarrow^*_\mathcal{A} a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, $a_j \in A$.

In both cases, let $U\langle t\rangle = Cl_\mathcal{C}(Q\langle t\rangle)$.

Induction step: Let $height(t) \geq 1$, and assume that for all trees $s \in T_\Sigma$ with $height(s) < height(t)$ we defined $U\langle s\rangle$. First we define $Q\langle t\rangle \in P_+(A)$ and then $U\langle t\rangle \in P_+(A)$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then $t = f(t_1, \ldots, t_m)$ for some $f \in \Sigma_m$, $m \geq 1$, and $t_1, \ldots, t_m \in T_\Sigma$. Then let

$$Q\langle t\rangle = \{ a \in A \mid f(a_1, \ldots, a_m) \rightarrow a \in R_\mathcal{A} \text{ and } a_1 \in U\langle t_1\rangle, ..., a_m \in U\langle t_m \rangle \}.$$

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $r \in CO_\Sigma(X_m)$ for some $m \geq 0$, and $t = r[t_1, \ldots, t_m]$ for some $t_1, \ldots, t_m \in T_\Sigma$. Then let $Q\langle t\rangle$ consist of all states $a \in A$ such that

(1) $r[a_1, \ldots, a_m] \rightarrow^*_\mathcal{A} a$ for some $a_1 \in W_1, ..., a_m \in W_m$, or

(2) there is a term rewriting rule $l \rightarrow r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

- $var(l) = X_h$ and $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{ x_{i_1}, \ldots, x_{i_k} \} = var(l)$ and
- $l'[a_1, \ldots, a_k] \rightarrow^*_\mathcal{A} a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in U\langle s_{i_j}\rangle$, otherwise $a_j \in A$.

In both cases, let $U\langle t\rangle = Cl_\mathcal{C}(Q\langle t\rangle)$.

Lemma 3.8 *Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ and $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$ be a connected total dbta. For every $t \in T_\Sigma$, $t \rightarrow^*_\mathcal{C} U\langle t\rangle$.*

Proof. We proceed by induction on $height(t)$.

Base of induction: $height(t) = 0$. Then $t = f$ for some $f \in \Sigma_0$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then by the definition of type 1 term rewriting rules of $R_{\mathcal{C}}$, $f \to Q\langle t\rangle$ is a type 1 term rewriting rule of $R_{\mathcal{C}}$. Accordingly $t \to_{\mathcal{C}} Q\langle t\rangle$.

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $t = r$ and by the definition of type 2 term rewriting rules of $R_{\mathcal{C}}$, $r \to Q\langle t\rangle$ is a type 2 term rewriting rule of $R_{\mathcal{C}}$. Consequently, $t \to_{\mathcal{C}} Q\langle t\rangle$.

In both cases, since $U\langle t\rangle = Cl_{\mathcal{C}}(Q\langle t\rangle)$, we have $Q\langle t\rangle \to_{\mathcal{C}}^{*} U\langle t\rangle$. Thus, $t \to_{\mathcal{C}} Q\langle t\rangle \to_{\mathcal{C}}^{*} U\langle t\rangle$.

Induction step: Let $height(t) \geq 1$, and assume that for all trees $s \in T_\Sigma$ with $height(s) < height(t)$ we showed that $s \to_{\mathcal{C}}^{*} U\langle s\rangle$. We now show that $t \to_{\mathcal{C}}^{*} U\langle t\rangle$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then $t = f(t_1, \ldots, t_m)$ for some $f \in \Sigma_m$ and $m \geq 1$. By the induction hypothesis, for each $i = 1, \ldots, m$, $t_i \to_{\mathcal{C}}^{*} U\langle t_i\rangle$. By the definition of type 1 term rewriting rules of $R_{\mathcal{C}}$, $f(U\langle t_1\rangle, \ldots, U\langle t_m\rangle) \to Q\langle t\rangle$ is a type 1 term rewriting rule of $R_{\mathcal{C}}$. Accordingly

$$t = f(t_1, \ldots, t_m) \underset{\mathcal{C}}{\overset{*}{\to}} f(U\langle t_1\rangle, \ldots, U\langle t_m\rangle) \underset{\mathcal{C}}{\to} Q\langle t\rangle.$$

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $r \in T_\Sigma(X_m)$, $var(r) = X_m$ for some $m \geq 0$ and $t = r[t_1, \ldots, t_m]$ for some $t_1, \ldots, t_m \in T_\Sigma$. By the induction hypothesis, for each $i = 1, \ldots, m$, $t_i \to_{\mathcal{C}}^{*} U\langle t_i\rangle$. By the definition of type 2 term rewriting rules of $R_{\mathcal{C}}$, $r[U\langle t_1\rangle, \ldots, U\langle t_m\rangle] \to Q\langle t\rangle$ is a type 2 term rewriting rule of $R_{\mathcal{C}}$. Accordingly

$$t = r[t_1, \ldots, t_m] \underset{\mathcal{C}}{\overset{*}{\to}} r[U\langle t_1\rangle, \ldots, U\langle t_m\rangle] \underset{\mathcal{C}}{\to} Q\langle t\rangle.$$

In both cases, as $U\langle t\rangle = Cl_{\mathcal{C}}(Q\langle t\rangle)$, we have $Q\langle t\rangle \to_{\mathcal{C}}^{*} U\langle t\rangle$. Therefore,

$$t \underset{\mathcal{C}}{\overset{*}{\to}} Q\langle t\rangle \underset{\mathcal{C}}{\overset{*}{\to}} U\langle t\rangle.$$

□

Lemma 3.9 *Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. For every $p, t \in T_\Sigma$, if an IO one-pass reduction sequence*

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_n, l_n \to r_n}{\to} s_n = t \text{ with } n \geq 0 \tag{7}$$

holds with R, then $p^{\mathcal{A}} \in U\langle t\rangle$.

Proof. We proceed by induction on the length n of (7).

Base of induction: $n = 0$. Then $p = t$. By the definition of $U\langle t\rangle$, $p^{\mathcal{A}} \in U\langle t\rangle$ holds.

Induction step: Let $n \geq 0$, and assume that for all integers less than or equal to n, the statement holds. We now show that the statement holds for $n + 1$ as well. Let

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_{n+1}, l_{n+1} \to r_{n+1}}{\to} s_{n+1} = t \text{ with } n \geq 0$$

be an arbitrary IO one-pass reduction sequence. Then

- $l_{n+1} \to r_{n+1}$ is a term rewriting rule in R for some $l_{n+1} \in T_\Sigma(X_h)$, $h \geq 0$, such that
  - $l_{n+1} = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{ x_{i_1}, \ldots, x_{i_k} \} = var(l_{n+1})$ and
  - $r_{n+1} \in rhs(R)$ and $r_{n+1} \in T_\Sigma(X_m)$, $var(r_{n+1}) = X_m$ for some $m \geq 0$,
- $s_n = u[l_{n+1}[t_1, \ldots, t_h]] = l'[t_{i_1}, \ldots, t_{i_k}]$ for some $t_1, \ldots, t_h \in T_\Sigma$,
- $t = u[r_{n+1}[t_1, \ldots, t_h]]$,
- $p = u[l'[p_1, \ldots, p_k]]$, and
- for every $j = 1, \ldots, k$, $p_{i_j} \Rightarrow_{R,IO,n_j} t_{i_j}$ with $n_j \leq n$.

Recall that, by the induction hypothesis, for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r_{n+1}) = X_m$, then $p_{i_j}^{\mathcal{A}} \in U\langle t_{i_j} \rangle$. By the definition of $Q\langle r_{n+1}[t_1, \ldots, t_h]\rangle$,

$$l'[p_1, \ldots, p_k]^{\mathcal{A}} \in Q\langle r_{n+1}[t_1, \ldots, t_h]\rangle.$$

Hence $l'[p_1, \ldots, p_k]^{\mathcal{A}} \in U\langle r_{n+1}[t_1, \ldots, t_h]\rangle$. By the definition of type 1 term rewriting rules of $R_{\mathcal{C}}$, by induction on $height(u)$, we can show that $u[l'[p_1, \ldots, p_k]^{\mathcal{A}}]^{\mathcal{A}} \in Q\langle t \rangle$ and $u[l'[p_1, \ldots, p_k]^{\mathcal{A}}]^{\mathcal{A}} \in U\langle t \rangle$. Hence $p^{\mathcal{A}} = u[l'[p_1, \ldots, p_k]^{\mathcal{A}}]^{\mathcal{A}} \in U\langle t \rangle$.

□

Lemmas 3.6, 3.8, and 3.9 imply the following result.

Lemma 3.10 *Let R be a right-linear knlv rwo TRS$_+$ over a ranked alphabet Σ and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. For any $t \in T_\Sigma$, $t \rightarrow_{\mathcal{C}}^* U\langle t \rangle$ and $\{\, p^{\mathcal{A}} \mid p \in T_\Sigma, p \Rightarrow_{R,IO} t \,\} = U\langle t \rangle$.*

Lemma 3.11 *Let R be a right-linear knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. Then $IOSF(L(\mathcal{A})) \subseteq L(\mathcal{C})$.*

Proof. Let $t \in IOSF(L(\mathcal{A}))$. Then there is $p \in L(\mathcal{A})$ such that $p \Rightarrow_{R,IO} t$, thus $p^{\mathcal{A}} \in A_f$. By Lemma 3.10, $t \rightarrow_{\mathcal{C}}^* U\langle t \rangle$ and $\{\, p^{\mathcal{A}} \mid p \in T_\Sigma, p \Rightarrow_{R,IO} t \,\} = U\langle t \rangle$. Hence $U\langle t \rangle$ is a final state of C. Consequently, $t \in L(\mathcal{C})$. □

Lemma 3.12 *Let R be a right-linear knlv rwo TRS$_+$ over a ranked alphabet Σ and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. Then $L(\mathcal{C}) \subseteq IOSF(L(\mathcal{A}))$.*

Proof. Let $t \in L(\mathcal{C})$, then there is $V \in C_f$ such that $t \rightarrow_{\mathcal{C}}^* V$. By the definition of C_f, we have $V \cap A_f \neq \emptyset$. Let $a \in V \cap A_f$, then by Lemma 3.6 there is $p \in T_\Sigma$ such that $p^{\mathcal{A}} = a$ and $p \Rightarrow_{R,IO} t$. Therefore, $p \in L(\mathcal{A})$ and $t \in IOSF(L(\mathcal{A}))$. □

Finally, we can prove Theorem 3.2.

Proof of Theorem 3.2. Lemma 3.11 and Lemma 3.12 imply the theorem. □

For all total dbtas $\mathcal{A}$ and $\mathcal{B}$ we can construct a dbta $\mathcal{C}$ such that $L(\mathcal{C}) = L(\mathcal{A}) \cap L(\mathcal{B})$, and for every total dbta $\mathcal{A}$, we can decide whether $L(\mathcal{A}) = \emptyset$. Thus, as a corollary to Theorem 3.2, we get that for right-linear knlv rwo TRS$_+$s the second-order IO one-pass joinability problem is decidable.

Corollary 3.13 *For any right-linear knlv rwo TRS$_+$ R and recognizable tree languages L and M over a ranked alphabet Σ,*

- *we can construct a total dbta $\mathcal{A}$ such that $L(\mathcal{A}) = IOSF(L) \cap IOSF(M)$, and*
- *it is decidable whether $IOSF(L) \cap IOSF(M) \neq \emptyset$.*

By Example 3.1, we observe that for a recognizable tree language L the set $IOSF(L)$ in general is not a recognizable tree language. However, when deciding whether $IOSF(L) \subseteq M$ for any recognizable tree language M, this does not cause a problem. This is because the inclusion $IOSF(L) \subseteq M$ does not imply that all elements of M should be in $IOSF(L)$. Therefore we do not have to decide whether $t \in IOSF(L)$ for every tree $t \in M$.

We now show that for knlv rwo TRS$_+$s the second-order IO one-pass inclusion problem and the second-order IO one-pass reachability problem are decidable.

Theorem 3.14 *For any right-linear knlv rwo TRS$_+$ R and recognizable tree languages L and M over a ranked alphabet Σ, it is decidable whether $IOSF(L) \subseteq M$ and whether $IOSF(L) \cap M \neq \emptyset$.*

We now outline the proof of the theorem. Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ, L and M be recognizable tree languages over Σ, and let the connected total dbtas $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be such that $L(\mathcal{A}) = L$ and $L(\mathcal{B}) = M$. We construct the gbta $\mathcal{D}$ by extending the construction of the gbta $\mathcal{C}$ appearing in the proof of Theorem 3.2; we add a second component to the state of $\mathcal{C}$, in which $\mathcal{D}$ mimics $\mathcal{B}$ on the output tree. Dropping the second components of the states of $\mathcal{D}$ we get back the gbta $\mathcal{C}$ of Theorem 3.2. We illustrate the gbta $\mathcal{D}$ by an example. Then we show Lemmas 3.16, 3.17, 3.18, and 3.19, which describe the connections between the IO one-pass reduction sequences of R, the dbtas $\mathcal{A}$, $\mathcal{B}$, and the gbta $\mathcal{D}$. Finally we show Lemma 3.20 and Lemma 3.21, which imply the decidability results of the theorem.

Intuitively, along an IO one-pass reduction sequence, several input subtrees p may be reduced by R to the same output tree t. Then an instance of a non-linear left-hand side may appear as a result of rewriting different input subtrees to the same tree. To solve the above problem, we construct a gbta $\mathcal{D}$ processing the IO one-pass sentential forms t, it simulates both $\mathcal{A}$ on all input trees p such that $p \Rightarrow_{R,IO} t$ and $\mathcal{B}$ on the sentential form t.

For any knlv rwo TRS$_+$ R and connected total dbtas $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$, we now introduce the gbta $\mathcal{D}(R, \mathcal{A}, \mathcal{B})$, denoted simply by $\mathcal{D}$ when R, $\mathcal{A}$, and $\mathcal{B}$ are understood from the context. Let $\mathcal{D} = (\Sigma, P_+(A) \times B, R_{\mathcal{D}}, \emptyset)$, where $R_{\mathcal{D}}$ consists of the following term rewriting rules of two types.

Type 1. For all $f \in \Sigma_m$ with $m \geq 0$, all $W_1, \ldots, W_m, W \in P_+(A)$ and $b_1, \ldots, b_m, b \in B$ such that

$$W = \{ a \in A \mid f(a_1, \ldots, a_m) \to a \in R_{\mathcal{A}} \text{ and } a_1 \in W_1, \ldots, a_m \in W_m \}$$

and

$$f(b_1, \ldots, b_m) \to b \in R_{\mathcal{B}},$$

let $R_{\mathcal{D}}$ contain the term rewriting rule $f((W_1, b_1), \ldots, (W_m, b_m)) \to (W, b)$.

Type 2. For every right-hand side $r \in rhs(R)$ with assuming that $r \in T_\Sigma(X_m)$, $var(r) = X_m$ for some $m \geq 0$, and for all $W_1, \ldots, W_m \in P_+(A)$, $b_1, \ldots, b_m, b \in B$, let $R_{\mathcal{D}}$ contain the term rewriting rule

$$r[(W_1, b_1), \ldots, (W_m, b_m)] \to (W, b), \tag{8}$$

where $r[b_1, \ldots, b_m] \to^*_{\mathcal{B}} b$, and W consists of all states $a \in A$ such that

(1) $r[a_1, \ldots, a_m] \to^*_{\mathcal{A}} a$ for some $a_1 \in W_1, ..., a_m \in W_m$, or

(2) there is a term rewriting rule $l \to r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

* $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $\{ x_{i_1}, \ldots, x_{i_k} \} = var(l)$ and
* $l'[a_1, \ldots, a_k] \to^*_{\mathcal{A}} a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in W_{i_j}$, otherwise $a_j \in A$.

Then we say that the right-hand side $r \in rhs(R)$ yields the term rewriting rule

$$r[(W_1, b_1), \ldots, (W_m, b_m)] \to (W, b).$$

Note that if r is a variable, that is $r = x_i$ for some $i \geq 1$, then the type 2 term rewriting rule

$$r[(W_1, b_1), \ldots, (W_m, b_m)] \to (W, b)$$

is of the form $(W_i, b_i) \to (W, b)$. By the definition of type 2 term rewriting rules, for each $(V, b) \in P_+(A) \times B$, there is a unique term rewriting rule $(V, b) \to (W, b)$ in $R_{\mathcal{D}}$ with left-hand side (V, b). Furthermore, for each term rewriting rule $(V, b) \to (W, b) \in R_{\mathcal{D}}$, we have $V \subseteq W$. Hence for each $(V, b) \in P_+(A) \times B$, there is a unique state $(Z, b) \in P_+(A) \times B$ such that $(V, b) \to^*_{\mathcal{D}} (Z, b)$ and $(Z, b) \to (Z, b) \in R_{\mathcal{D}}$. We call (Z, b) the $\mathcal{D}$ closure of (V, b) and denote (Z, b) by $Cl_{\mathcal{D}}(V, b)$.

Example 3.15 Let the ranked alphabet Σ, the recognizable tree language L, and the dbta $\mathcal{A}$ be the same as in Example 3.4, and let $M = L$ and $\mathcal{B} = \mathcal{A}$. Consider the knlv rwo TRS$_+$

$$R = \{ f(x_1, x_2) \to \#,\ f(f(x_1, x_1), x_2) \to f(x_1, \$) \}$$

over Σ. Then $\mathcal{D} = (\Sigma, P_+(\{ 0, 1 \}) \times \{ 0, 1 \}, R_{\mathcal{D}}, \emptyset)$, where $R_{\mathcal{D}}$ consists of the following term rewriting rules:

Type 1 term rewriting rules:

$$\# \to (\{ 1 \}, 1),\ \$ \to (\{ 0 \}, 0),$$

$$f((\{0\},0),(\{0\},0)) \to (\{0\},0), f((\{0\},0),(\{0\},1)) \to (\{0\},1),$$
$$f((\{0\},1),(\{0\},0)) \to (\{0\},1), f((\{0\},1),(\{0\},1)) \to (\{0\},0),$$
$$f((\{0\},0),(\{1\},0)) \to (\{1\},0), f((\{0\},0),(\{1\},1)) \to (\{1\},1),$$
$$f((\{0\},1),(\{1\},0)) \to (\{1\},1), f((\{0\},1),(\{1\},1)) \to (\{1\},0),$$
$$f((\{0\},0),(\{0,1\},0)) \to (\{0,1\},0), f((\{0\},0),(\{0,1\},1)) \to (\{0,1\},1),$$
$$f((\{0\},1),(\{0,1\},0)) \to (\{0,1\},1), f((\{0\},1),(\{0,1\},1)) \to (\{0,1\},0),$$
$$\dots$$
$$f((\{0,1\},0),(\{0,1\},0)) \to (\{0,1\},0), f((\{0,1\},0),(\{0,1\},1)) \to (\{0,1\},1),$$
$$f((\{0,1\},1),(\{0,1\},0)) \to (\{0,1\},1), f((\{0,1\},1),(\{0,1\},1)) \to (\{0,1\},0),$$

Type 2 term rewriting rules:

$$\# \to (\{1\},1), f(\{0\},0),\$) \to (\{0,1\},0),\ f(\{0\},1),\$) \to (\{0,1\},0),$$
$$f(\{1\},0),\$) \to (\{0,1\},0),\ f(\{1\},1),\$) \to (\{0,1\},0),$$
$$f(\{0,1\},0),\$) \to (\{0,1\},0),\ f(\{0,1\},1),\$) \to (\{0,1\},0).$$

Lemma 3.16 *Let R be a right-linear knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be connected total dbtas. For every $t \in T_\Sigma$, $V \in P_+(A)$, and $b_0 \in B$, if a reduction sequence*

$$t = s_0 \underset{\mathcal{D}}{\to} s_1 \underset{\mathcal{D}}{\to} \cdots \underset{\mathcal{D}}{\to} s_n = (V, b_0) \text{ with } n \geq 0 \tag{9}$$

holds, then $t^{\mathcal{B}} = b_0$ and for every $a_0 \in V$, there is $p \in T_\Sigma$ such that $p^{\mathcal{A}} = a_0$ and $p \Rightarrow_{R,IO} t$.

Proof. We proceed by induction on the number ν of applications of type 2 term rewriting rules along the reduction sequence (9).

Base of induction: $\nu = 0$. By the definition of type 1 term rewriting rules of $R_{\mathcal{D}}$, by induction on n we can show that $V = \{ t^{\mathcal{A}} \}$ and $t^{\mathcal{B}} = b_0$. Let $p = t$. Obviously, $p^{\mathcal{A}} = t^{\mathcal{A}}$ and $p \Rightarrow_{R,IO} t$.

Induction step: Let $\nu \geq 0$, and assume that for all integers less than or equal to ν, the statement holds. We now show that the statement holds for $\nu + 1$ as well. Let $0 \leq j \leq n-1$ be such that

- in the $j+1$th step $s_j \to_R s_{j+1}$ of (9), $\mathcal{D}$ applies a type 2 term rewriting rule

  $$r[(W_1, b_1), \ldots, (W_m, b_m)] \to (W, b) \in R_{\mathcal{D}}$$

  yielded by some right-hand side $r \in rhs(R)$, where

  - $m \geq 0$ and $r \in CO_\Sigma(X_m)$,
  - $(W_1, b_1), \ldots, (W_m, b_m), (W, b) \in P_+(A) \times B$, and

- along the last $n - j - 1$ steps $s_{j+1} \to_{\mathcal{D}} s_{j+2} \to_{\mathcal{D}} \cdots \to_{\mathcal{D}} s_n$ of (9), $\mathcal{D}$ applies only term rewriting rules of type 1.

For each $i = 1, \ldots, m$, let $\alpha_i \in POS(r)$ such that $lab(r, \alpha_i) = x_i$. Then by Statement 2.8, there exist $u \in CO_\Sigma(X_{\ell+1})$ and $t_1, \ldots, t_m, v_1, \ldots, v_\ell \in T_\Sigma$ with $m, \ell \geq 0$ such that

- $t = u[r[t_1, \ldots, t_m], v_1, \ldots, v_\ell]$
- $s_j = u[r[(W_1, b_1) \ldots, (W_m, b_m)], (V_1, d_1), \ldots, (V_\ell, d_\ell)]$,
- for each $i = 1, \ldots, m$, the restriction of (9) to $adr(u, x_1)\alpha_i$ is of the form

  $$t_i = z_{i0} \to_{\mathcal{D}} z_{i1} \to_{\mathcal{D}} \cdots \to_{\mathcal{D}} z_{ik_i} = (W_i, b_i)$$

  for some $k_i \geq 0$ and $z_{i\eta} \in T_{\Sigma \cup (P_+(A) \times B)}$ for $\eta = 1, \ldots, k_i$.

- for each $i = 1, \ldots, \ell$, the restriction of (9) to $adr(u, x_{i+1})$ is of the form

  $$v_i = w_{i0} \to_{\mathcal{D}} w_{i1} \to_{\mathcal{D}} \cdots \to_{\mathcal{D}} w_{i\vartheta_i} = (V_i, d_i)$$

  for some $\vartheta_i \geq 0$ and $w_{i\eta} \in T_{\Sigma \cup (P_+(A) \times B)}$ for $\eta = 1, \ldots, \vartheta_i$.
- $s_{j+1} = u[(W, b), (V_1, d_1), \ldots, (V_\ell, d_\ell)]$.

By the definition of type 1 term rewriting rules of $R_{\mathcal{D}}$, by induction on $n - j - 1$ we can show that

$$V = \{\, a' \mid u[a, c_1, \ldots, c_\ell] \xrightarrow[\mathcal{A}]{*} a' \text{ for some } a \in W, c_1 \in V_1, \ldots, c_\ell \in V_\ell \,\}$$

and

$$u[b, d_1, \ldots, d_\ell] \xrightarrow[\mathcal{B}]{*} b_0.$$

Hence

$u[a, c_1, \ldots, c_\ell] \to_{\mathcal{A}}^{*} a_0$ for some $a \in W$ and $c_1 \in V_1, \ldots, c_\ell \in V_\ell$.

Recall that $r \in rhs(R)$ with assuming that $r \in CO_\Sigma(X_m)$ for some $m \geq 0$, yields the type 2 term rewriting rule $r[(W_1, b_1), \ldots, (W_m, b_m)] \to (W, b) \in R_{\mathcal{D}}$. By the definition of type 2 term rewriting rules of $R_{\mathcal{D}}$,

$$r[b_1, \ldots, b_m] \xrightarrow[\mathcal{B}]{*} b$$

and W consists of all states $a \in A$ such that Condition (1) or Condition (2) holds.

First assume that Condition (1) in the definition of type 2 term rewriting rules holds for a. That is to say, $r[a_1, \ldots, a_m] \to_{\mathcal{A}}^{*} a$ for some $a_1 \in W_1, ..., a_m \in W_m$. Then by the induction hypothesis,

for every $i = 1, \ldots, m$, there is $p_i \in T_\Sigma$ such that $p_i^{\mathcal{A}} = a_i$ and $p_i \Rightarrow_{R,IO} t_i$.

Again by the induction hypothesis,

for every $j = 1, \ldots, \ell$, there is $q_j \in T_\Sigma$ such that $q_j^{\mathcal{A}} = c_j$ and $q_j \Rightarrow_{R,IO} v_j$.

Then let $p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell]$. Therefore

- $t = u[r[t_1, \ldots, t_m], v_1, \ldots, v_\ell] \to_{\mathcal{B}}^{*} u[r[b_1, \ldots, b_m], d_1, \ldots, d_\ell] \to_{\mathcal{A}}^{*} u[b, d_1, \ldots, d_\ell] \to_{\mathcal{A}}^{*} b_0$,
- $p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell] \to_{\mathcal{A}}^{*} u[r[a_1, \ldots, a_m], c_1, \ldots, c_\ell] \to_{\mathcal{A}}^{*} u[a, c_1, \ldots, c_\ell] \to_{\mathcal{A}}^{*} a_0$, and
- there is an IO one-pass reduction sequence

  $$p = u[r[p_1, \ldots, p_m], q_1, \ldots, q_\ell] = z_0 \xrightarrow[R]{} z_1 \xrightarrow[R]{} \cdots \xrightarrow[R]{} u[r[t_1, \ldots, t_h], v_1, \ldots, v_\ell] = t.$$

Second, assume that Condition (2) in the definition of type 2 term rewriting rules holds for a. That is, there is a term rewriting rule $l \to r \in R$ for some $l \in T_\Sigma(X_h)$, $h \geq 0$, such that

- $var(l) = X_h$ and $l = l'[x_{i_1}, \ldots, x_{i_k}]$ for some $l' \in CO_\Sigma(X_k)$, $k \geq 0$, and $x_{i_1}, \ldots, x_{i_k} \in X_h$ and
- $l'[a_1, \ldots, a_k] \to_{\mathcal{A}}^{*} a$ for some $a_1, ..., a_k$ such that for every $j = 1, \ldots, k$, if $x_{i_j} \in var(r)$ then $a_j \in W_{i_j}$, otherwise $a_j \in A$.

For every $j = 1, \ldots, k$, we distinguish two cases.

Case 1: $x_{i_j} \in var(r)$. Then $a_j \in W_{i_j}$. Hence by the induction hypothesis, there is $p_j \in T_\Sigma$ such that $p_j^{\mathcal{A}} = a_j$ and $p_j \Rightarrow_{R,IO} t_{i_j}$.

Case 2: $x_{i_j} \notin var(r)$. Then let $p_j \in T_\Sigma$ be arbitrary.

Since R is knlv, if x_{i_j} is a non-linear variable of l, then $x_{i_j} \in var(r)$. Accordingly

$$l'[p_1, \ldots, p_k] \underset{R,IO}{\Rightarrow} l'[t_{i_1}, \ldots, t_{i_k}] = l[t_1, \ldots, t_h].$$

Again by the induction hypothesis,

for every $j = 1, \dots, \ell$, there is $q_j \in T_\Sigma$ such that $q_j^{\mathcal{A}} = c_j$ and $q_j \Rightarrow_{R,IO} v_j$.

Then let

$$p = u[l'[p_1, \dots, p_k], q_1, \dots, q_\ell].$$

Therefore

- $p = u[l'[p_1, \dots, p_k], q_1, \dots, q_\ell] \to_{\mathcal{A}}^* u[l'[a_1, \dots, a_k], c_1, \dots, c_\ell] \to_{\mathcal{A}}^* u[a, c_1, \dots, c_\ell] \to_{\mathcal{A}}^* a_0$, and
- there is an IO one-pass reduction sequence

$$p = u[l'[p_1, \dots, p_k], q_1, \dots, q_\ell] = s_0 \underset{R}{\to} s_1 \underset{R}{\to} \cdots \underset{R}{\to} u[l'[t_{i_1}, \dots, t_{i_k}], v_1, \dots, v_\ell] =$$

$$u[l[t_1, \dots, t_h], v_1, \dots, v_\ell] \underset{R}{\to} u[r[t_1, \dots, t_m], v_1, \dots, v_\ell] = t.$$

□

As in the proof of Lemma 3.6, in the last step of the above proof we used our assumption that R is right-linear and hence $r \in rhs(R)$ is linear as well.

Recall that in Definition 3.7 we defined the sets $Q\langle t, R, \mathcal{A}\rangle \in P_+(A)$ and $U\langle t, R, \mathcal{A}\rangle \in P_+(A)$, denoted simply as $Q\langle t\rangle$ and $U\langle t\rangle$ respectively when R and $\mathcal{A}$ are understood from the context.

Lemma 3.17 *Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be connected total dbtas. For every $t \in T_\Sigma$, $t \to_{\mathcal{D}}^* (U\langle t\rangle, t^{\mathcal{B}})$.*

Proof. We proceed by induction on $height(t)$.

Base of induction: $height(t) = 0$. Then $t = f$ for some $f \in \Sigma_0$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then by the definition of type 1 term rewriting rules of $R_{\mathcal{D}}$, $f \to (Q\langle t\rangle, t^{\mathcal{B}})$ is a type 1 term rewriting rule of $R_{\mathcal{D}}$. Hence $t \to_{\mathcal{D}} (U\langle t\rangle, t^{\mathcal{B}})$.

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $t = r$ and by the definition of type 2 term rewriting rules of $R_{\mathcal{D}}$, $r \to (Q\langle t\rangle, r^{\mathcal{B}})$ is a type 2 term rewriting rule of $R_{\mathcal{D}}$. Consequently, $r \to_{\mathcal{D}} (Q\langle t\rangle, r^{\mathcal{B}})$.

In both cases, as $U\langle t\rangle = Cl_{\mathcal{D}}(Q\langle t\rangle)$, we have $(Q\langle t\rangle, r^{\mathcal{B}}) \to_{\mathcal{D}}^* (U\langle t\rangle, r^{\mathcal{B}})$. Therefore

$$t \underset{\mathcal{D}}{\to} (Q\langle t\rangle, r^{\mathcal{B}}) \overset{*}{\underset{\mathcal{D}}{\to}} (U\langle t\rangle, r^{\mathcal{B}}).$$

Induction step: Let $height(t) \geq 1$, and assume that for all trees s with $height(s) < height(t)$ we showed that $s \to_{\mathcal{D}}^* (U\langle s\rangle, s^{\mathcal{B}})$. We now show that $t \to_{\mathcal{D}}^* (U\langle t\rangle, t^{\mathcal{B}})$. We distinguish two cases.

Case 1: t is not an instance of a right-hand side $r \in rhs(R)$. Then $t = f(t_1, \dots, t_m)$ for some $f \in \Sigma_m$ and $m \geq 1$. By the induction hypothesis, for each $i = 1, \dots, m$, $t_i \to_{\mathcal{D}}^* (U\langle t_i\rangle, t_i^{\mathcal{B}})$. By the definition of type 1 term rewriting rules of $R_{\mathcal{D}}$, $f((U\langle t_1\rangle, t_1^{\mathcal{B}}) \dots, (U\langle t_m\rangle, t_m^{\mathcal{B}})) \to Q\langle t\rangle$ is a type 1 term rewriting rule of $R_{\mathcal{D}}$. Accordingly

$$t = f(t_1, \dots, t_m) \overset{*}{\underset{\mathcal{D}}{\to}} f((V\langle t_1\rangle, t_1^{\mathcal{B}}), \dots, (U\langle t_m\rangle, t_m^{\mathcal{B}})) \underset{\mathcal{D}}{\to} (Q\langle t\rangle, t^{\mathcal{B}}).$$

Case 2: t is an instance of a right-hand side $r \in rhs(R)$. Then $r \in T_\Sigma(X_m)$, $var(r) = X_m$ for some $m \geq 0$ and $t = r[t_1, \dots, t_m]$ for some $t_1, \dots, t_m \in T_\Sigma$. By the induction hypothesis, for each $i = 1, \dots, m$, $t_i \to_{\mathcal{D}}^* (U\langle t_i\rangle, t_i^{\mathcal{B}})$. By the definition of type 2 term rewriting rules of $R_{\mathcal{D}}$, $r[(U\langle t_1\rangle, t_1^{\mathcal{B}}), \dots, (U\langle t_m\rangle, t_m^{\mathcal{B}})] \to (Q\langle t\rangle, r[t_1^{\mathcal{B}}, \dots, t_m^{\mathcal{B}}]^{\mathcal{B}})$ is a type 2 term rewriting rule of $R_{\mathcal{D}}$. Observe that $t^{\mathcal{B}} = r[t_1^{\mathcal{B}}, \dots, t_m^{\mathcal{B}}]^{\mathcal{B}}$. Accordingly

$$t = r[t_1, \dots, t_m] \overset{*}{\underset{\mathcal{D}}{\to}} r[(U\langle t_1\rangle, t_1^{\mathcal{B}}), \dots, (U\langle t_m\rangle, t_m^{\mathcal{B}})] \underset{\mathcal{D}}{\to} (Q\langle t\rangle, t^{\mathcal{B}}).$$

In both cases, as $U\langle t\rangle = Cl_{\mathcal{D}}(Q\langle t\rangle)$, we have $(Q\langle t\rangle, t^{\mathcal{B}}) \to_{\mathcal{D}}^* (U\langle t\rangle, t^{\mathcal{B}})$. Therefore,

$$t \underset{\mathcal{D}}{\to} (Q\langle t\rangle, t^{\mathcal{B}}) \overset{*}{\underset{\mathcal{D}}{\to}} (U\langle t\rangle, t^{\mathcal{B}}).$$

□

Lemma 3.18 *Let R be a knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be connected total dbtas. For every $p, t \in T_\Sigma$, if an IO one-pass reduction sequence*

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_{n+1}, l_{n+1} \to r_{n+1}}{\to} s_n = t \text{ with } n \geq 0 \tag{10}$$

holds with R, then $t \to^*_{\mathcal{D}} (U\langle t\rangle, t^{\mathcal{B}})$ and $p^{\mathcal{A}} \in U\langle t\rangle$.

Proof. By Lemma 3.17, $t \to^*_{\mathcal{D}} (U\langle t\rangle, t^{\mathcal{B}})$. By Lemma 3.19, $p^{\mathcal{A}} \in U\langle t\rangle$. □

Lemmas 3.16, 3.17, and 3.18 imply the following result.

Lemma 3.19 *Let R be a right-linear knlv rwo TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be connected total dbtas. For any $t \in T_\Sigma$, $t \to^*_{\mathcal{D}} (U\langle t\rangle, t^{\mathcal{B}})$ and $\{ p^{\mathcal{A}} \mid p \in T_\Sigma, p \Rightarrow_{R,IO} t \} = U\langle t\rangle$.*

Lemma 3.20 *Let R be a right-linear knlv rwo TRS, L and M be recognizable tree languages over a ranked alphabet Σ. The following are equivalent:*

- *$IOSF(L) \subseteq M$, and*
- *for every $(W, b) \in REACH_{\mathcal{D}}$, if $W \cap A_f \neq \emptyset$, then $b \in B_f$.*

Proof. First, assume that $IOSF(L) \subseteq M$. Take an arbitrary element (W, b) of $REACH_{\mathcal{D}}$ such that $W \cap A_f \neq \emptyset$, and let $a_0 \in W \cap A_f$ be arbitrary. Then there is $t \in T_\Sigma$ such that $t \to^*_{\mathcal{D}} (W, b)$. By Lemma 3.16, $t^{\mathcal{B}} = b$ and there is $p \in T_\Sigma$ such that $p^{\mathcal{A}} = a_0$ and $p \Rightarrow_{R,IO} t$. Consequently $p \in L$ and hence $t \in IOSF(L)$. By our assumption, $t \in M$. Since $t^{\mathcal{B}} = b$, we have $b \in B_f$. Recall that (W, b) is an arbitrary element of $REACH_{\mathcal{D}}$ such that $W \cap A_f \neq \emptyset$. Thus for every $(W, b) \in REACH_{\mathcal{D}}$, if $W \cap A_f \neq \emptyset$, then $b \in B_f$.

Second, assume that for every $(W, b) \in REACH_{\mathcal{D}}$, if $W \cap A_f \neq \emptyset$, then $b \in B_f$. Let $t \in IOSF(L)$ be arbitrary. Then there is $p \in L$ such that $p \Rightarrow_{R,IO} t$. By Lemma 3.18, there is $V \in P_+(A)$ such that

$$t \to^*_{\mathcal{D}} (V, t^{\mathcal{B}}) \text{ and } p^{\mathcal{A}} \in V.$$

Then $(V, t^{\mathcal{B}}) \in REACH_{\mathcal{D}}$. Furthermore, as $p \in L$, we have $p^{\mathcal{A}} \in A_f$. Consequently, by our assumption, $t^{\mathcal{B}} \in B_f$, and thus $t \in M$. Since $t \in IOSF(L)$ is arbitrary, we have $IOSF(L) \subseteq M$. □

Lemma 3.21 *Let R be a right-linear knlv rwo TRS, L and M be recognizable tree languages over a ranked alphabet Σ. $IOSF(L) \cap M \neq \emptyset$ if and only if there is $(W, b) \in REACH_{\mathcal{D}}$ such that $W \cap A_f \neq \emptyset$ and $b \in B_f$.*

Proof. First assume that $IOSF(L) \cap M \neq \emptyset$. Then there are $p \in L$ and $t \in M$ such that $p \Rightarrow_{R,IO} t$. Then $p^{\mathcal{A}} \in A_f$ and $t^{\mathcal{B}} \in B_f$. By Lemma 3.17,

$$t \to^*_{\mathcal{D}} (W, t^{\mathcal{B}}), \text{ where } W = \{ q^{\mathcal{A}} \mid q \in T_\Sigma \text{ and } q \Rightarrow_{R,IO} t \}.$$

Consequently, $p^{\mathcal{A}} \in W$, and thus $p^{\mathcal{A}} \in W \cap A_f$. Therefore $W \cap A_f \neq \emptyset$. Obviously, $(W, t^{\mathcal{B}}) \in REACH_{\mathcal{D}}$, hence we are done.

Second, assume that there is $(W, b) \in REACH_{\mathcal{D}}$ such that $W \cap A_f \neq \emptyset$, and $b \in B_f$. Then there is $t \in T_\Sigma$ such that $t \to^*_{\mathcal{D}} (W, b)$. Let $a_0 \in W \cap A_f$ be arbitrary. By Lemma 3.16, $t^{\mathcal{B}} = b$ and there is $p \in T_\Sigma$ such that $p^{\mathcal{A}} = a_0$ and $p \Rightarrow_{R,IO} t$. Consequently, $t \in M$, $p \in L$, and $t \in IOSF(L)$. Therefore, $t \in IOSF(L) \cap M$ implying that $IOSF(L) \cap M \neq \emptyset$. □

Finally, we can prove Theorem 3.14.

Proof of Theorem 3.14. Lemma 3.20 and Lemma 3.21 imply the theorem. □

For any left-linear TRS$_+$ R over Σ and recognizable tree language L, given a connected total dbta $\mathcal{A}$ over Σ recognizing L, we construct a gbta $\mathcal{E}$ recognizing the set $\{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R,IO} t \}$. On an input tree p, $\mathcal{E}$ simulates the computation of $\mathcal{A}$ on some t where $p \Rightarrow_{R,IO} t$.

Theorem 3.22 *For any left-linear TRS$_+$ R and recognizable tree language L over a ranked alphabet Σ, we can construct a gbta $\mathcal{E}$ over Σ such that $L(\mathcal{E}) = \{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R,IO} t \}$.*

Let R be a left-linear TRS$_+$ over a ranked alphabet Σ, L be recognizable tree language over Σ, and let the connected total dbta $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$ be such that $L(\mathcal{A}) = L$. We now outline the proof of the theorem. We construct a gbta $\mathcal{E}$ over Σ, and illustrate it by an example. Then we show a series of lemmas, Lemmas 3.24 and 3.25 describe the connections between the IO one-pass reduction sequences of R, the dbta $\mathcal{A}$, and the gbta $\mathcal{E}$. Finally we show Lemma 3.26 and Lemma 3.27, which imply $L(\mathcal{E}) = \{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R,IO} t \}$.

Without loss of generality we may assume that from now on throughout this section, for every term rewriting rule $l \to r \in R$ there are integers $m, \mu \geq 0$ such that $l \in CO_\Sigma(X_m)$ and $r \in T_\Sigma(X_{m+\mu})$, $var(r) \subseteq X_{m+\mu}$, $x_{m+1}, \ldots, x_{m+\mu} \in var(r)$. For any left-linear TRS$_+$ R over a ranked alphabet Σ, and connected total dbta $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$, we now introduce the gbta $\mathcal{E}(R, \mathcal{A})$, denoted simply by $\mathcal{E}$ when R and $\mathcal{A}$ are understood from the context. Let $\mathcal{E} = (\Sigma, A, R_\mathcal{E}, A_f)$, where $R_\mathcal{E}$ consists of the following term rewriting rules of two types.

- Type 1. $R_\mathcal{A} \subseteq R_\mathcal{E}$.
- Type 2. For every term rewriting rule $l \to r \in R$ with assuming that there are integers $m, \mu \geq 0$ such that
  - $l \in CO_\Sigma(X_m)$ and
  - $r \in T_\Sigma(X_{m+\mu})$, $var(r) \subseteq X_{m+\mu}$, and $x_{m+1}, \ldots, x_{m+\mu} \in var(r)$,

  and for all states $a_1, \ldots, a_m, a_{m+1}, \ldots, a_{m+\mu}, a \in A$ such that $r[a_1, \ldots, a_{m+\mu}] \to^*_\mathcal{A} a$,
  let $R_\mathcal{E}$ contain the term rewriting rule

  $$l[a_1, \ldots, a_m] \to a.$$

  Here we say that the term rewriting rule $l \to r$ yields the term rewriting rule $l[a_1, \ldots, a_m] \to a$.

Example 3.23 Let the ranked alphabet Σ, the recognizable tree language L, and the dbta $\mathcal{A}$ be the same as Example 3.4. Consider the TRS$_+$

$$R = \{ f(x_1, x_2) \to \#,\ f(f(x_1, x_2), \#) \to f(x_1, f(x_3, x_3)) \}$$

over Σ. Then $\mathcal{E} = (\Sigma, \{0, 1\}, R_\mathcal{E}, \{0\})$, where $R_\mathcal{E}$ consists of the following term rewriting rules of two types.
Type 1 term rewriting rules:

$$\# \to 1,$$

$$f(0,0) \to 0,\ f(0,1) \to 1,\ f(1,0) \to 1,\ f(1,1) \to 0.$$

Type 2 term rewriting rules:

$$f(0,0) \to 1,\ f(0,1) \to 1,\ f(1,0) \to 1,\ f(1,1) \to 1,$$

$$f(f(0,0),\#) \to 0,\ f(f(0,1),\#) \to 0,\ f(f(1,0),\#) \to 1,\ f(f(1,1),\#) \to 1.$$

An input p of $\mathcal{E}$ is the input of an IO one-pass reduction sequence of $\mathcal{R}$ with output t. Computing on p, $\mathcal{E}$ mimics some reduction steps of $\mathcal{A}$ on p and also the computation of $\mathcal{A}$ on the right-hand sides of the applied term rewriting rules along the IO one-pass reduction sequence of R. At each node of p, $\mathcal{E}$ has a choice

- to consider this node as a node of p and to mimic $\mathcal{A}$ on p, or
- to attempt to recognize the left-hand side l of a term rewriting rule $l \to r \in R$, and in case of success to mimic $\mathcal{A}$ on the right-hand side r. For each non-linear variable x_i in the right-hand side r, if $\mathcal{E}$ arrives at x_i in the left-hand side l in state a_i, then we substitute the state a_i of $\mathcal{A}$ for all occurrences of x_i in r.

We now compare the gbta $\mathcal{E}$ with the gbta $\mathcal{D}$ in the proof of Theorem 3.14. Observe that the TRS$_+$ R is left-linear, hence it cannot happen that along an IO one-pass reduction sequence an instance of a non-linear left-hand side may appear as a result of rewriting different input subtrees to the same tree. Hence we do not need to compute for any tree $t \in T_\Sigma$ the state set $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t \}$. Consequently, the states of $\mathcal{E}$ are the states of A rather than state sets in $P_+(A)$. Therefore, we do not need the rwo property of R, which in the proof of Theorem 3.14 enabled us to compute the set $\{ p^{\mathcal{A}} \mid p \Rightarrow_{R,IO} t \}$.

Example 3.1 shows that in general, $IOSF(L)$ is not a recognizable tree language. However this does not cause a problem, because $IOSF(L)$ is quite different from the tree language recognized by $\mathcal{E}$, since $L(\mathcal{E})$ consists of trees which are reduced by R to the elements of L.

We now prove a series of lemmas.

Lemma 3.24 *Let R be a left-linear TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. For every $p, t \in T_\Sigma$, if an IO one-pass reduction sequence*

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_n, l_n \to r_n}{\to} s_n = t \text{ with } n \geq 0 \tag{11}$$

holds with R, then $p \to_{\mathcal{E}}^* t^{\mathcal{A}}$.

Proof. We proceed by induction on the length n of (11).

Base of induction: $n = 0$. Then $p = t$. By the definition of the type 1 term rewriting rules of $R_{\mathcal{E}}$, by induction on $height(p)$ we can show that $p \to_{\mathcal{E}}^* p^{\mathcal{A}}$.

Induction step: Let $n \geq 0$, and assume that for all integers less than or equal to n, the statement holds. We now show that the statement holds for $n + 1$ as well. Let

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\to} s_1 \underset{\alpha_2, l_2 \to r_2}{\to} s_2 \underset{\alpha_3, l_3 \to r_3}{\to} \cdots \underset{\alpha_{n+1}, l_{n+1} \to r_{n+1}}{\to} s_{n+1} = t$$

be an IO one-pass reduction sequence. Then

- the term rewriting rule $l_{n+1} \to r_{n+1}$ is in R with $l_{n+1} \in CO_\Sigma(X_m)$ and $r_{n+1} \in T_\Sigma(X_k)$ for some $m \geq 0$ and $k \geq m$,
- $p = u[l_{n+1}[p_1, \ldots, p_m], v_1, \ldots, v_\ell]$ for some $u \in CO_\Sigma(X_{\ell+1})$, $\ell \geq 0$, $p_1, \ldots, p_m \in T_\Sigma$, $v_1, \ldots, v_\ell \in T_\Sigma$,
- $s_n = u[l_{n+1}[t_1, \ldots, t_m], w_1, \ldots, w_\ell]$ for some $t_1, \ldots, t_m \in T_\Sigma$ and $w_1, \ldots, w_\ell \in T_\Sigma$, where
  - for every $i = 1, \ldots, m$, $p_i \Rightarrow_{R,IO,n_i} t_i$ with $n_i \leq n$, and
  - for every $i = 1, \ldots, \ell$, $v_i \Rightarrow_{R,IO,\nu_i} w_i$ with $\nu_i \leq n$, and
- $t = u[r_{n+1}[t_1, \ldots, t_m, z_1, \ldots, z_\mu], w_1, \ldots, w_\ell]$ for some $z_1, \ldots, z_\mu \in T_\Sigma$.

Then

$$r_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \ldots, z_\mu^{\mathcal{A}}] \xrightarrow[\mathcal{A}]{*} r_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \ldots, z_\mu^{\mathcal{A}}]^{\mathcal{A}},$$

and

$$u[r_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \ldots, z_\mu^{\mathcal{A}}]^{\mathcal{A}}, w_1, \ldots, w_\ell] \xrightarrow[\mathcal{A}]{*} t^{\mathcal{A}}.$$

By the induction hypothesis

$$\text{for every } i = 1, \ldots, m, p_i \xrightarrow[\mathcal{E}]{*} t_i^{\mathcal{A}}.$$

$$\text{for every } i = 1, \ldots, \ell, v_i \xrightarrow[\mathcal{E}]{*} w_i^{\mathcal{A}}.$$

By the construction of type 2 term rewriting rules of $R_{\mathcal{E}}$,

$$l_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}] \to r_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \ldots, z_\mu^{\mathcal{A}}]^{\mathcal{A}} \in R_{\mathcal{E}}$$

and by the construction of type 1 term rewriting rules of $R_{\mathcal{E}}$,

$$u[r_{n+1}[t_1^{\mathcal{A}}, \ldots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \ldots, z_\mu^{\mathcal{A}}]^{\mathcal{A}}, w_1^{\mathcal{A}}, \ldots, w_\ell^{\mathcal{A}}] \xrightarrow[\mathcal{E}]{*} t^{\mathcal{A}}.$$

Thus

$$p = u[l_{n+1}[p_1, \dots, p_m], v_1, \dots, v_\ell] \to^*_{\mathcal{E}} u[l_{n+1}[t_1^{\mathcal{A}}, \dots, t_m^{\mathcal{A}}], w_1^{\mathcal{A}}, \dots, w_\ell^{\mathcal{A}}] \to_{\mathcal{E}}$$

$$u[r_{n+1}[t_1^{\mathcal{A}}, \dots, t_m^{\mathcal{A}}, z_1^{\mathcal{A}}, \dots, z_\mu^{\mathcal{A}}]^{\mathcal{A}}, w_1^{\mathcal{A}}, \dots, w_\ell^{\mathcal{A}}] \to^*_{\mathcal{E}} t^{\mathcal{A}}.$$

□

Lemma 3.25 *Let R be a left-linear TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. For every $p \in T_\Sigma$ and $a \in A$, if a reduction sequence*

$$p = s_0 \underset{\mathcal{E}}{\to} s_1 \underset{\mathcal{E}}{\to} \cdots \underset{\mathcal{E}}{\to} s_n = a, \text{ with } n \geq 0 \tag{12}$$

holds, then there is $q \in T_\Sigma$ such that $q^{\mathcal{A}} = a$ and $p \Rightarrow_{R,IO} q$.

Proof. We proceed by induction on the number ν of applications of type 2 term rewriting rules along (12).

Base of induction: $\nu = 0$. Then let $q = p$. By the definition of type 1 term rewriting rules of $R_{\mathcal{E}}$, by induction on n we can show that $p \to^*_{\mathcal{A}} a$. Obviously, $p \Rightarrow_{R,IO} p$.

Induction step: Let $\nu \geq 0$, and assume that for all integers less than or equal to ν, the statement holds. We now show that the statement holds for $\nu + 1$ as well. Let $0 \leq j \leq n-1$ be such that

- in the $j+1$th step $s_j \to_R s_{j+1}$ of (12), $\mathcal{E}$ applies a type 2 term rewriting rule $l[a, \dots, a_m] \to a \in R_{\mathcal{E}}$ yielded by some term rewriting rule $l \to r \in R$, where $l, r \in T_\Sigma(X_m)$, $m \geq 0$, $a_1, \dots, a_m, a \in A$, and
- along the last $n-j-1$ steps $s_{j+1} \to_{\mathcal{E}} s_{j+2} \to_{\mathcal{E}} \cdots \to_{\mathcal{E}} s_n$ of (12) $\mathcal{E}$ applies only term rewriting rules of type 1.

Then by Statement 2.8, there exist $u \in CO_\Sigma(X_{\ell+1})$ and $p_1, \dots, p_m, v_1, \dots, v_\ell \in T_\Sigma$ with $m, \ell \geq 0$ such that

- $p = u[l[p_1, \dots, p_m], v_1, \dots, v_\ell]$,
- $s_j = u[l[a_1, \dots, a_m], b_1, \dots, b_\ell]$ for some $b_1, \dots, b_\ell \in A$,
- for each $i = 1, \dots, m$, the restriction of (12) to $adr(u, x_1)adr(l, x_i)$ is of the form

  $$p_i = z_{i0} \to_{\mathcal{E}} z_{i1} \to_{\mathcal{E}} \cdots \to_{\mathcal{E}} z_{ik_i} = a_i$$

  for some $k_i \geq 0$ and $z_{i1}, \cdots z_{ik_i} \in T_{\Sigma \cup A}$,
- for each $i = 1, \dots, \ell$, the restriction of (12) to $adr(u, x_{i+1})$ is of the form

  $$v_i = w_{i0} \to_{\mathcal{E}} w_{i1} \to_{\mathcal{E}} \cdots \to_{\mathcal{E}} w_{i\vartheta_i} = b_i$$

  for some $\vartheta_i \geq 0$ and $w_{i1}, \cdots, w_{i\vartheta_i} \in T_{\Sigma \cup A}$,
- $s_{j+1} = u[a, b_1, \dots, b_\ell]$, and
- $s_n = a_0$.

Recall that along the last $n-j-1$ steps $s_{j+1} \to_{\mathcal{E}} s_{j+2} \to_{\mathcal{E}} \cdots \to_{\mathcal{E}} s_n$ of (12) $\mathcal{E}$ applies only term rewriting rules of type 1. By the definition of type 1 term rewriting rules of $R_{\mathcal{E}}$, by induction on $n-j-1$ we can show that

$$u[a, b_1, \dots, b_\ell] \underset{\mathcal{A}}{\overset{*}{\to}} a_0.$$

Recall that the term rewriting rule $l \to r \in R$ yields the type 2 term rewriting rule

$$l[a, \dots, a_m] \to a \in R_{\mathcal{E}}.$$

By the definition of type 2 term rewriting rules of $R_{\mathcal{E}}$, there are $a_{m+1}, \dots a_{m+\mu} \in A$ such that

$$r[a_1, \dots, a_m, a_{m+1}, \dots a_{m+\mu}] \xrightarrow[\mathcal{A}]{*} a.$$

By the induction hypothesis, for every $i = 1, \dots, m$, there is $q_i \in T_\Sigma$ such that

$$q_i \xrightarrow[\mathcal{A}]{*} a_i \text{ and } p_i \underset{R,IO}{\Rightarrow} q_i.$$

Since $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ is a connected total dbta, there are $q_{m+1}, \dots, q_{m+\mu} \in T_\Sigma$ such that for every $i = 1, \dots, \mu$,

$$q_{m+i} \xrightarrow[\mathcal{A}]{*} a_{m+i}.$$

Again by the induction hypothesis, for every $i = 1, \dots, \ell$, there is $w_i \in T_\Sigma$ such that

$$w_i \xrightarrow[\mathcal{A}]{*} b_i \text{ and } v_i \underset{R,IO}{\Rightarrow} w_i.$$

Then let $q = u[r[q_1, \dots, q_m, q_{m+1}, \dots, q_{m+\mu}], w_1, \dots, w_\ell]$. Consequently

$$\begin{aligned} &q = u[r[q_1, \dots, q_m, q_{m+1}, \dots, q_{m+\mu}], w_1, \dots, w_\ell] \to_{\mathcal{A}}^* \\ &u[r[a_1, \dots, a_m, a_{m+1}, \dots, a_{m+\mu}], b_1, \dots, b_\ell] \to_{\mathcal{A}} u[a, b_1, \dots, b_\ell] \to_{\mathcal{A}}^* a_0, \text{ and} \end{aligned}$$

there is an IO one-pass reduction sequence

$$\begin{aligned} &p = u[l[p_1, \dots, p_m], v_1, \dots, v_\ell] = s_0 \to_R s_1 \to_R \dots \to_R u[l[q_1, \dots, q_m], w_1, \dots, w_\ell]] \to_R \\ &u[r[q_1, \dots, q_m, q_{m+1}, \dots, q_{m+\mu}]] = q. \end{aligned}$$

□

Lemma 3.26 *Let R be a left-linear TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. Then $\{ p \in T_\Sigma \mid \exists t \in L(\mathcal{A}).\, p \Rightarrow_{R,IO} t \} \subseteq L(\mathcal{E})$.*

Proof. Let $p \in T_\Sigma$ and assume that there is $t \in L$ such that $p \Rightarrow_{R,IO} t$. Then $t^{\mathcal{A}} \in A_f$. By Lemma 3.24, $p \to_{\mathcal{E}}^* t^{\mathcal{A}}$. Consequently, $p \in L(\mathcal{E})$. □

Lemma 3.27 *Let R be a left-linear TRS$_+$ over a ranked alphabet Σ, and $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be a connected total dbta. Then $L(\mathcal{E}) \subseteq \{ p \in T_\Sigma \mid \exists q \in L(\mathcal{A}).\, p \Rightarrow_{R,IO} q \}$.*

Proof. Let $p \in L(\mathcal{E})$. Then $p \to_{\mathcal{E}}^* a$ for some $a \in A_f$. By Lemma 3.25, there is $q \in T_\Sigma$ such that $q^{\mathcal{A}} = a$ and $p \Rightarrow_{R,IO} q$. Therefore, $q \in L(\mathcal{A})$. □

Finally, we can prove Theorem 3.22.

Proof of Theorem 3.22. Lemma 3.26 and Lemma 3.27 imply the theorem. □

We now show that for left-linear TRS$_+$s, the second-order IO one-pass common ground ancestor problem is decidable.

Corollary 3.28 *For any left-linear TRS$_+$ R and recognizable tree languages L and M over a ranked alphabet Σ, it is decidable whether there is a term $t \in T_\Sigma$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$.*

Proof. By Theorem 3.22, we construct total dbtas $\mathcal{E}$ and $\mathcal{D}$ such that

$$L(\mathcal{E}) = \{ p \in T_\Sigma \mid \exists t \in L.\, p \Rightarrow_{R,IO} t \}$$

and

$$L(\mathcal{D}) = \{ p \in T_\Sigma \mid \exists t \in M.\, p \Rightarrow_{R,IO} t \}.$$

Then we construct a total dbta $\mathcal{F}$ such that $L(\mathcal{F}) = L(\mathcal{E}) \cap L(\mathcal{D})$, and decide whether $L(\mathcal{F}) \neq \emptyset$. The the answer is yes if and only if there is a term $t \in T_\Sigma$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$. $\square$

By Proposition 2.7 for left-linear TRS_+s, the second-order IO one-pass common ancestor problem is decidable.

Corollary 3.29 *For any left-linear TRS_+ R and recognizable tree languages L and M over a ranked alphabet Σ, it is decidable whether there is a term $t \in T_\Sigma(X)$ such that $IOSF(t) \cap L \neq \emptyset$ and $IOSF(t) \cap M \neq \emptyset$.*

4 OI One-Pass Reductions

We show that for left-linear wa TRSs, the second-order OI one-pass common ancestor problem is decidable. Consider the left-linear wa TRS R and the recognizable tree language L in Example 3.1, we noted that $OISF(L)$ is not a recognizable tree language. Although $OISF(L)$ is not recognizable in general, for any left-linear wa TRS R and recognizable tree language L over Σ, we can construct a gbta $\mathcal{B}$ such that $L(\mathcal{B}) = \{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R,OI} t \}$. This is because $L(\mathcal{B})$ is different from both $OISF(L)$ and the set of those trees $p \in T_\Sigma$ such that $p \Rightarrow_{R,IO} t$ for some $t \in OISF(L)$.

Statement 4.1 *If R is a left-linear wa TRS, then R^{-1} is a right-linear knlv rwo TRS_+.*

Proof. By direct inspection of the definitions. $\square$

Lemma 4.2 *For any left-linear wa TRS R over a ranked alphabet Σ and $p, t \in T_\Sigma$,*

$p \Rightarrow_{R,OI} t$ *if and only if* $t \Rightarrow_{R^{-1},IO} p$.

Proof. ($\Rightarrow$) Assume that $p \Rightarrow_{R,OI} t$, i.e. there is an OI one-pass reduction sequence

$$p = s_0 \underset{\beta_1, l_1 \to r_1; \alpha_1}{\to} s_1 \underset{\beta_2, l_2 \to r_2; \alpha_2}{\to} s_2 \underset{\beta_3, l_3 \to r_3; \alpha_3}{\to} \cdots \underset{\beta_n, l_n \to r_n; \alpha_n}{\to} s_n = t \text{ with } n \geq 0 \tag{13}$$

for R. Then by induction on the length n of (13), we show that $t \Rightarrow_{R^{-1},IO} p$.

Base of induction: $n = 0$. Then $p = t$ and hence $t \Rightarrow_{R^{-1},IO} p$.

Induction step: Let $n \geq 0$, and assume that for all integers less than or equal to n, the statement holds. We now show that the statement holds for $n+1$ as well. Let

$$p = s_0 \underset{\beta_1, l_1 \to r_1; \alpha_1}{\to} s_1 \underset{\beta_2, l_2 \to r_2; \alpha_2}{\to} s_2 \underset{\beta_3, l_3 \to r_3; \alpha_3}{\to} \cdots \underset{\beta_{n+1}, l_{n+1} \to r_{n+1}; \alpha_{n+1}}{\to} s_{n+1} = t \text{ with } n \geq 0 \tag{14}$$

be an OI one-pass reduction sequence for R. By (14) we have

- $l_1 \to r_1 \in R$ with $l_1 \in CO_\Sigma(X_m)$ and $r_1 \in T_\Sigma(X_m)$ for some $m \geq 0$,
- $p = u[l_1[p_1, \ldots, p_m]]$ for some $u \in CO_\Sigma(X_1)$, $p_1, \ldots, p_m \in T_\Sigma$,
- $s_1 = u[r_1[t_1, \ldots, t_m]]$, where for every $i = 1, \ldots, m$, $t_i \in T_\Sigma$ and $p_i \Rightarrow_{R,OI,n_i} t_i$ with $n_i \leq n$, and
- $t = u[r_1[t_1, \ldots, t_m]]$.

Since the term rewriting rule $l_1 \to r_1$ is in R, the term rewriting rule $r_1 \to l_1$ is in R^{-1}. By the induction hypothesis, for every $i = 1, \ldots, m$, there is an IO one-pass reduction sequence

$$t_i = s_{i0} \to_{R^{-1},IO} s_{i1} \to_{R^{-1},IO} \cdots \to_{R^{-1},IO} s_{in_i} = p_i$$

with R^{-1} for some $n_i \geq 0$ and $s_{i0}, s_{i1}, \ldots, s_{in_i} \in T_\Sigma$. Consequently, we have the IO one-pass reduction sequence

$$t = u[r_1[t_1, \ldots, t_m]] \to_{R^{-1},IO} u[r_1[s_{11}, t_2, \ldots, t_m]] \to_{R^{-1},IO}$$

$u[r_1[s_{12}, t_2, \ldots, t_m]] \rightarrow_{R^{-1},IO} \cdots \rightarrow_{R^{-1},IO} u[r_1[p_1, t_2, \ldots, t_m]] \rightarrow_{R^{-1},IO} \cdots$

$\rightarrow_{R^{-1},IO} u[r_1[p_1, \ldots, p_{m-1}, t_m]] \rightarrow_{R^{-1},IO} u[r_1[p_1, \ldots, p_{m-1}, s_{m1}]] \rightarrow_{R^{-1},IO}$

$u[r_1[p_1, \ldots, p_{m-1}, s_{m2}]] \rightarrow_{R^{-1},IO} \cdots \rightarrow_{R^{-1},IO} u[r_1[p_1, \ldots, p_m]] \rightarrow_{R^{-1},IO} u[l_1[p_1, \ldots, p_m]] =$
p

with R.

($\Leftarrow$) Assume that $p \Rightarrow_{R^{-1},IO} t$, i.e., there is an IO one-pass reduction sequence

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\rightarrow} s_1 \underset{\alpha_2, l_2 \to r_2}{\rightarrow} s_2 \underset{\alpha_3, l_3 \to r_3}{\rightarrow} \cdots \underset{\alpha_n, l_n \to r_n}{\rightarrow} s_n = t \text{ with } n \geq 0. \tag{15}$$

Then by induction on the length n of (15), we show that $t \Rightarrow_{R,OI} p$.

Base of induction: $n = 0$. Then $p = t$ and hence $t \Rightarrow_{R,OI} p$.

Induction step: Let $n \geq 0$, and assume that for all integers less than or equal to n, the statement holds. We now show that the statement holds for $n + 1$ as well. Let

$$p = s_0 \underset{\alpha_1, l_1 \to r_1}{\rightarrow} s_1 \underset{\alpha_2, l_2 \to r_2}{\rightarrow} s_2 \underset{\alpha_3, l_3 \to r_3}{\rightarrow} \cdots \underset{\alpha_{n+1}, l_{n+1} \to r_{n+1}}{\rightarrow} s_{n+1} = t \tag{16}$$

be an IO one-pass reduction sequence with R^{-1}. Then

- there is a term rewriting rule $l_{n+1} \to r_{n+1} \in R^{-1}$ with $l_{n+1} \in T_\Sigma(X_m)$ and $r_{n+1} \in CO_\Sigma(X_m)$ for some $m \geq 0$,
- $p = u[l_{n+1}[p_1, \ldots, p_m]]$ for some $u \in CO_\Sigma(X_1)$, $p_1, \ldots, p_m \in T_\Sigma$,
- $s_n = u[l_{n+1}[t_1, \ldots, t_m]]$ for some $t_1, \ldots, t_m \in T_\Sigma$,
- for every $i = 1, \ldots, m$, $p_i \Rightarrow_{R,OI,n_i} t_i$ with $n_i \leq n$, and
- $t = u[r_{n+1}[t_1, \ldots, t_m]]$.

Since the term rewriting rule $l_{n+1} \to r_{n+1}$ is in R^{-1}, the term rewriting rule $r_{n+1} \to l_{n+1}$ is in R. By the induction hypothesis, for every $i = 1, \ldots, m$, there is an OI one-pass reduction sequence

$t_i = s_{i0} \rightarrow_{R^{-1},OI} s_{i1} \rightarrow_{R^{-1},OI} s_{i2} \rightarrow_{R^{-1},OI} \cdots \rightarrow_{R^{-1},OI} s_{i\nu_i} = p_i$

with R^{-1} for some $\nu_i \geq 0$ and $s_{i0}, s_{i1}, \ldots, s_{i\nu_i} \in T_\Sigma$. Consequently, we have the OI one-pass reduction sequence

$t = u[r_{n+1}[t_1, \ldots, t_m]] \rightarrow_{R,OI} u[l_{n+1}[t_1, \ldots, t_m]] \rightarrow_{R,OI} u[l_{n+1}[s_{11}, t_2, \ldots, t_m]] \rightarrow_{R,OI}$

$u[l_{n+1}[s_{12}, t_2, \ldots, t_m]] \rightarrow_{R,OI} \cdots \rightarrow_{R,OI} u[l_{n+1}[p_1, t_2, \ldots, t_m]] \rightarrow_{R,OI} \cdots$

$\rightarrow_{R,OI} u[l_{n+1}[p_1, \ldots, p_{m-1}, t_m]] \rightarrow_{R,OI} u[l_{n+1}[p_1, \ldots, p_{m-1}, s_{m1}]] \rightarrow_{R,OI}$

$u[l_{n+1}[p_1, \ldots, p_{m-1}, s_{m2}]] \rightarrow_{R,OI} \cdots \rightarrow_{R,OI} u[l_{n+1}[p_1, \ldots, p_m]] = p$

with R.

Theorem 4.3 *For any left-linear wa TRS R and recognizable tree language L over a ranked alphabet Σ, we can construct a gbta $\mathcal{B}$ such that $L(\mathcal{F}) = \{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R,OI} t \}$.*

Proof. Let R be a left-linear wa TRS and L be a recognizable tree language over a ranked alphabet Σ. By Statement 4.1, R^{-1} is a right-linear knlv rwo TRS$_+$. Consequently, by Theorem 3.2, we can construct a gbta $\mathcal{F}$ such that $L(\mathcal{F}) = \{ p \in T_\Sigma \mid \exists t \in L.\ p \Rightarrow_{R^{-1},IO} t \}$. By Lemma 4.2 for all $p, t \in T_\Sigma$, $p \Rightarrow_{R,OI} t$ if and only if $t \Rightarrow_{R^{-1},IO} p$. Hence $L(\mathcal{F}) = \{ p \in T_\Sigma \mid \exists t \in L.\ t \Rightarrow_{R,OI} t \}$.
$\square$

We now show that for left-linear wa TRSs, the second-order OI one-pass common ground ancestor problem is decidable.

Corollary 4.4 *For any left-linear wa TRS$_+$ R and recognizable tree languages L and M over a ranked alphabet Σ, it is decidable whether there is a term $p \in T_\Sigma$ such that $OISF(p) \cap L \neq \emptyset$ and $OISF(p) \cap M \neq \emptyset$.*

Proof. By Theorem 4.3, we construct total dbtas $\mathcal{F}$ and $\mathcal{G}$ such that

$$L(\mathcal{F}) = \{\, p \in T_\Sigma \mid \exists t \in L.\, p \Rightarrow_{R,OI} t \,\}$$

and

$$L(\mathcal{G}) = \{\, p \in T_\Sigma \mid \exists t \in M.\, p \Rightarrow_{R,OI} t \,\}.$$

We construct a total dbta $\mathcal{H}$ such that $L(\mathcal{H}) = L(\mathcal{F}) \cap L(\mathcal{G})$. Then we decide whether $L(\mathcal{H}) \neq \emptyset$. The answer is yes if and only if there is a term $p \in T_\Sigma$ such that $OISF(p) \cap L \neq \emptyset$ and $OISF(p) \cap M \neq \emptyset$. $\square$

By Proposition 2.7 and Corollary 4.4 for left-linear wa TRSs, the second-order OI one-pass common ancestor problem is decidable.

Corollary 4.5 *For any left-linear wa TRS$_+$ R and recognizable tree languages L and M over a ranked alphabet Σ, it is decidable whether there is a term $t \in T_\Sigma(X)$ such that $OISF(t) \cap L \neq \emptyset$ and $OISF(t) \cap M \neq \emptyset$.*

5 Conclusion

We now describe some related work. Fülöp et al. [7] considered two very restrictive strategies of term rewriting: one-pass root-started rewriting and one-pass leaf-started rewriting. When we follow the former startegy, rewriting starts at the root of the input term and proceeds continuously towards the leaves. We do not rewrite any part of the current term obtained in a previous rewriting step. When no more rewriting is possible, a one-pass root-started normal form of the original term has been reached. Such a normal form may be reducible in the ususal sense, as a rewriting term rewriting rule may apply either in the part already rewritten or to a subtree which was not rewritten. The leaf-started version is similar, but the rewriting is initiated at the leaves and proceeds towards the root. The requirement that rewriting always concerns positions immediately adjacent to parts of the term rewritten in previous steps distinguishes these two rewriting strategies from the IO and OI one-pass reductions for TRSs [4, 19].

We constructed gbtas and dbtas recognizing various tree languages associated with the one-pass reductions for TRS$_+$s and recognizable tree languages. We now compare these constructions with the construction of tree automata that recognize descendants of initial terms where the construction is based on iteration of computing a tree automaton from a previous one [16, 17, 11]. Thus the construction is done in stepwise manner: given an initial tree automaton $\mathcal{A}_0$, then they construct a series of tree automata $\mathcal{A}_1, \mathcal{A}_2, \ldots$ until $\mathcal{A}_k = \mathcal{A}_{k+1}$. On the other hand, in this paper we construct in one step the resulting gbta or dbta, we do not iterate computing a tree automaton from a previous one. This is because of the very definition of a one-pass reduction sequence. For example, if along an IO one-pass reduction sequence we reduce by the term rewriting rule $l \to r$, then we never reduce the instance of the right-hand side r for the applied substitution.

We collected together the known results in the literature in Table 1, and then summed up the known results together with our contribution and some open problems in Table 2. The following problems remain open.

Problem 5.1 *Generalize the results of the paper to more general classes of term rewriting systems.*

Problem 5.2 *Given a TRS R and a recognizable tree language L, is it decidable whether $IOSF(L)$ is recognizable and whether $OISF(L)$ is recognizable?*

Gilleron and Tison [19] noted that for any linear TRS$_+$ R, $\Rightarrow_{R,IO}$ is equal to $\Rightarrow_{R,OI}$. Hence for any linear TRS$_+$ R and recognizable tree language L, $IOSF(L) = OISF(L)$. Therefore, we raise the following question.

Problem 5.3 *Given a TRS R and a recognizable tree language L, is it decidable whether $IOSF(L) \subseteq OISF(L)$, whether $OISF(L) \subseteq IOSF(L)$, and whether $IOSF(L) = OISF(L)$?*

References

[1] Baader, F. and Nipkow, T. Term Rewriting and All That, Cambridge University Press, Cambridge, United Kingdom, 1998.

[2] Bojanczyk, Hoffman. P., Reachability in Unions of Commutative Rewriting Systems Is Decidable, in: W. Thomas, P. Weil eds. STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings. Lecture Notes in Computer Science 4393, Springer 2007, ISBN 978-3-540-70917-6 pages 622-633

[3] Bojanczyk, M., Lasota, S., and Potapov, I. Reachability Problems - 9th International Workshop, RP 2015, Warsaw, Poland, September 21-23, 2015, Proceedings. Lecture Notes in Computer Science 9328, Springer 2015, ISBN 978-3-319-24536-2

[4] Dauchet, M., De Comité, F. A Gap Between Linear and Non Linear Term-Rewriting Systems. In Lescanne, P., editor, Rewriting Techniques and Applications, RTA-87, Proceedings. Lecture Notes in Computer Science 256 Springer 1987, pages 95-104.

[5] Feuillade, G., Genet, T. Reachability in Conditional Term Rewriting Systems. Electr. Notes Theor. Comput. Sci. 86(1): 133-146 (2003)

[6] Feuillade, G., Genet, T., Tong, V. V. T., Reachability Analysis over Term Rewriting Systems, Journal of Automated Reasoning **33** (2004) 341-383.

[7] Fülöp, Z., Jurvanen, E., Steinby, M., and Vágvölgyi, S. On one-pass term rewriting. In Brim, L., Gruska, J., Zlatuska, J. editors, Mathematical Foundations of Computer Science, 1998 Lecture Notes in Computer Science, **1450**, Springer Publishing Company, Berlin, 1998, pages 248-256; see also in Acta Cybernetica, **14** (1999) 83-98.

[8] Gécseg, F. and Steinby, M. *Tree Automata* (Akadémiai Kiadó, Budapest, 1984), online edition https://arxiv.org/abs/1509.06233 (2015).

[9] Gécseg, F. and Steinby, M. Tree languages. In G. Rozenberg and A. Salomaa, editors, Beyond Words, volume 3 of Handbook of Formal Languages, chapter 1, pages 1-68. Springer-Verlag, Berlin, 1997.

[10] Genet, T. Reachability analysis of rewriting for software verification, Habilitation document, 2009.
http://www.irisa.fr/celtique/genet/Publications/habilitation.pdf

[11] Genet, T. Termination criteria for tree automata completion. J. Log. Algebr. Meth. Program. 85 (2016) 3-33.

[12] Genet, T., Salmon Y., Reachability Analysis of Innermost Rewriting. Logical Methods in Computer Science **13** 2017

[13] Gilleron, R., Tison, S., Regular Tree Languages and Rewrite Systems. Fundamenta Informaticae **24** (1995) 157-174.

[14] Godoy G., Huntingford E., Innermost-Reachability and Innermost-Joinability Are Decidable for Shallow Term Rewrite Systems, In: Franz Baader F., editor Term Rewriting and Applications, 18th International Conference, RTA 2007, Proceedings, Lecture Notes in Computer Science 4533, Springer 2007, ISBN 978-3-540-73447-5 pages 184-199.

[15] Jacquemard F., Reachability and confluence are undecidable for flat term rewriting systems, Inf. Process. Lett. **87** (2003) 265-270.

[16] Jacquemard, F., Decidable Approximations of Term Rewriting Systems. In: Ganzinger, H., editor, Rewriting Techniques and Applications, 7th International Conference, RTA-96, 1996, Proceedings. Lecture Notes in Computer Science 1103, Springer 1996, ISBN 3-540-61464-8, pages 362-376.

[17] Jacquemard, F., Kojima, Y., and Sakai, M. Term Rewriting with Prefix Context Constraints and Bottom-Up Strategies, In: Felty, A.P., Middeldorp, A., editors, Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Proceedings. Lecture Notes in Computer Science 9195, Springer 2015, ISBN 978-3-319-21400-9 pages 137-151.

[18] Larsen, K. G., Potapov, I., and Srba, J. (Eds.) Reachability Problems - 10th International Workshop, RP 2016, Aalborg, Denmark, September 19-21, 2016, Proceedings. Lecture Notes in Computer Science 9899, Springer 2016, ISBN 978-3-319-45993-6

[19] Seynhaeve, F., Tison, S., Tommasi, M., Homomorphisms and Concurrent Term Rewriting. In Ciobanu, G., Paun, G. editors Fundamentals of Computation Theory, 12th International Symposium, FCT '99, Iasi, Romania, 1999, Proceedings. Springer 1999 Lecture Notes in Computer Science **1684** (1999) pages 475-487.

[20] Ouaknine, J., Potapov I., and Worrell, J., (Eds.) Reachability Problems - 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings. Lecture Notes in Computer Science 8762, Springer 2014, ISBN 978-3-319-11438-5

[21] Takai, T., Kaji, Y., and Seki, H., Right-Linear Finite Path Overlapping Term Rewriting Systems Effectively Preserve Recognizability, Rewriting Techniques and Applications, 11th International Conference, RTA 2000, Proceedings, (ed. L. Bachmair) Lecture Notes in Computer Science 1833, Springer Publishing Company, Berlin, 2000, 246-260.

[22] Tiwari, A., Approximate Reachability for Linear Systems, In: Oded Maler, Amir Pnueli editors, Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003, Proceedings. Lecture Notes in Computer Science 2623, Springer 2003, ISBN 3-540-00913-2, pages 514-525.

[23] Vágvölgyi, S., One-Pass Reductions, Acta Cybernetica, **22** (2016) 633-655.